



(12) 发明专利申请

(10) 申请公布号 CN 116795435 A

(43) 申请公布日 2023. 09. 22

(21) 申请号 202210261901.0

(22) 申请日 2022.03.15

(71) 申请人 华为技术有限公司

地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 万承臻 余涛 裴太乙 刘勋 张勇智 瞿时万

(74) 专利代理机构 广州三环专利商标代理有限公司 44202

专利代理师 张月婷

(51) Int. Cl.

G06F 9/4401 (2018.01)

权利要求书3页 说明书23页 附图16页

(54) 发明名称

兼容性管控方法及相关设备

(57) 摘要

本申请公开了一种兼容性管控方法及相关设备,在实现了使用统一的OS来支持所有类型的设备这一目标的情况下,在OS弹性部署到设备后,可以基于产品兼容性标识PCID和要求产品兼容性标识RPCID来判断应用是否可以在设备上安装并运行,若是,则可以向该设备分发该应用,从而可以解决设备与应用之间的兼容性问题。

设备兼容性标识	设备兼容性标识	应用兼容性标识	应用兼容性标识	用户设备
<p>S701: 根据OSID、应用兼容性标识PCID集合、设备兼容性标识RPCID集合和兼容性标识RPCID集合,生成兼容性标识RPCID集合。</p> <p>S702: 根据兼容性标识RPCID集合,生成兼容性标识RPCID集合。</p> <p>S703: 根据兼容性标识RPCID集合,生成兼容性标识RPCID集合。</p>	<p>S704: 设备兼容性标识RPCID集合。</p> <p>S705: 兼容性标识RPCID集合。</p> <p>S706: 兼容性标识RPCID集合。</p>	<p>S707: 兼容性标识RPCID集合。</p> <p>S708: 兼容性标识RPCID集合。</p> <p>S709: 兼容性标识RPCID集合。</p>	<p>S710: 兼容性标识RPCID集合。</p> <p>S711: 兼容性标识RPCID集合。</p> <p>S712: 兼容性标识RPCID集合。</p>	<p>S713: 兼容性标识RPCID集合。</p> <p>S714: 兼容性标识RPCID集合。</p> <p>S715: 兼容性标识RPCID集合。</p>
<p>S716: 兼容性标识RPCID集合。</p> <p>S717: 兼容性标识RPCID集合。</p> <p>S718: 兼容性标识RPCID集合。</p>	<p>S719: 兼容性标识RPCID集合。</p> <p>S720: 兼容性标识RPCID集合。</p> <p>S721: 兼容性标识RPCID集合。</p>	<p>S722: 兼容性标识RPCID集合。</p> <p>S723: 兼容性标识RPCID集合。</p> <p>S724: 兼容性标识RPCID集合。</p>	<p>S725: 兼容性标识RPCID集合。</p> <p>S726: 兼容性标识RPCID集合。</p> <p>S727: 兼容性标识RPCID集合。</p>	<p>S728: 兼容性标识RPCID集合。</p> <p>S729: 兼容性标识RPCID集合。</p> <p>S730: 兼容性标识RPCID集合。</p>

1. 一种通信系统,其特征在于,所述通信系统包括开发平台、应用分发平台、用户设备,其中:

所述开发平台用于:获取一个或多个部件,并基于所述一个或多个部件生成第一标识;其中,所述第一标识用于指示第一设备系统能力SysCap集合,所述第一设备SysCap集合是所述用户设备的操作系统的SysCap集合;

获取第二标识;其中,所述第二标识用于指示第一应用SysCap集合,所述第一应用SysCap集合是第一应用在所述用户设备上被安装并运行时所需要的SysCap集合;

所述应用分发平台用于:接收所述开发平台发送的所述第二标识;

获取所述第一标识;

在基于所述第一标识和所述第二标识确定所述第一设备SysCap集合包括所述第一应用SysCap集合的情况下,向所述用户设备发送第一请求,所述第一请求用于指示所述用户设备安装所述第一应用;

所述用户设备用于:安装所述第一应用。

2. 根据权利要求1所述的通信系统,其特征在于,所述第一标识为第一产品兼容性标识PCID,所述第二标识为第一要求的产品兼容性标识RPCID。

3. 根据权利要求1或2所述的通信系统,其特征在于,所述通信系统还包括设备认证平台,所述设备认证平台用于:

接收并保存所述开发平台发送的所述第一标识;

向所述应用分发平台发送所述第一标识。

4. 一种兼容性管控方法,应用于应用分发平台,其特征在于,所述方法包括:

所述应用分发平台获取第一标识,所述第一标识用于指示第一设备系统能力SysCap集合,所述第一设备SysCap集合是用户设备的操作系统的SysCap集合;

所述应用分发平台获取第二标识,所述第二标识用于指示第一应用SysCap集合,所述第一应用SysCap集合是第一应用在所述用户设备上被安装并运行时所需要的SysCap集合;

所述应用分发平台在基于所述第一标识和所述第二标识确定所述第一设备SysCap集合包括所述第一应用SysCap集合的情况下,向所述用户设备发送第一请求,所述第一请求用于指示所述用户设备安装所述第一应用。

5. 根据权利要求4所述的方法,其特征在于,所述第一标识为第一产品兼容性标识PCID,所述第二标识为第一要求的产品兼容性标识RPCID。

6. 根据权利要求5所述的方法,其特征在于,所述应用分发平台获取第一标识,具体包括:

所述应用分发平台向设备认证平台获取所述第一PCID;

其中,所述第一PCID是开发平台基于所述第一设备SysCap集合生成并发送给所述设备认证平台的。

7. 根据权利要求5或6所述的方法,其特征在于,所述应用分发平台获取第二标识,具体包括:

所述应用分发平台接收开发平台发送的第一RPCID;

其中,所述第一RPCID是所述开发平台基于所述第一应用SysCap集合生成的,所述第一应用SysCap集合是所述应用分发平台基于软件开发工具包SDK发布平台上发布的第一SDK

或所述第一PCID生成的。

8. 一种兼容性管控方法,应用于开发平台,其特征在于,所述方法包括:

所述开发平台获取第一信息,所述第一信息包括第一软件开发工具包SDK或第一标识,所述第一标识用于指示第一设备系统能力SysCap集合,所述第一设备SysCap集合是所述用户设备的操作系统的SysCap集合;

所述开发平台基于所述第一信息生成第二标识,所述第二标识用于指示第一应用SysCap集合,所述第一应用SysCap集合是第一应用在用户设备上被安装并运行时所需要的SysCap集合;

所述开发平台向应用分发平台发送所述第二标识。

9. 根据权利要求8所述的方法,其特征在于,所述第一标识为第一产品兼容性标识PCID,所述第二标识为第一要求的产品兼容性标识RPCID。

10. 根据权利要求9所述的方法,其特征在于,所述开发平台基于所述第一信息生成第二标识,具体包括:

在所述用户设备的设备类型为第一类型的情况下,所述开发平台基于所述第一SDK生成所述第一RPCID;

或,

在所述用户设备的设备类型不为所述第一类型的情况下,所述开发平台基于所述第一PCID生成所述第一RPCID。

11. 根据权利要求8-10任一项所述的方法,其特征在于,在所述开发平台获取第一信息之前,所述方法还包括:

所述开发平台获取一个或多个部件,并基于所述一个或多个部件生成所述第一标识。

12. 根据权利要求11所述的方法,其特征在于,所述开发平台基于所述一个或多个部件生成所述第一标识,具体包括:

所述开发平台基于所述一个或多个部件生成所述第一设备SysCap集合;

所述开发平台对所述第一设备SysCap集合进行编码生成所述第一PCID。

13. 根据权利要求11或12所述的方法,其特征在于,在所述开发平台基于所述一个或多个部件生成第一标识之后,所述方法还包括:

所述开发平台向设备认证平台发送第二请求,所述第二请求中包括所述第一PCID;

在所述设备认证平台基于所述第一PCID确认设备认证通过的情况下,所述开发平台接收并保存所述设备认证平台发送的第一令牌Token。

14. 一种兼容性管控方法,应用于用户设备,其特征在于,所述方法包括:

所述用户设备接收应用分发平台发送的第一请求,所述第一请求中包括第一要求的产品兼容性标识RPCID;

在所述用户设备基于所述第一RPCID和第一产品兼容性标识PCID确定第一设备SysCap集合包括所述第一应用SysCap集合的情况下,所述用户设备安装第一应用;

其中,所述第一RPCID用于指示第一应用SysCap集合,所述第一应用SysCap集合是所述第一应用在所述用户设备上被安装并运行时所需要的SysCap集合,所述第一PCID用于指示第一设备系统能力SysCap集合,所述第一设备SysCap集合是所述用户设备的操作系统的SysCap集合。

15. 根据权利要求14所述的方法,其特征在于,所述用户设备包括应用包管理服务PMS和Samgr部件,所述用户设备基于所述第一RPCID和第一产品兼容性标识PCID确定第一设备SysCap集合包括所述第一应用SysCap集合,具体包括:

所述用户设备利用所述PMS对所述第一RPCID进行解码生成所述第一应用SysCap集合;

所述用户设备利用所述Samgr部件查询所述第一设备SysCap集合;

所述用户设备利用所述PMS确定所述第一设备SysCap集合包括所述第一应用SysCap集合。

16. 根据权利要求14或15所述的方法,其特征在于,在所述用户设备安装第一应用之前,所述方法还包括:

所述用户设备显示第一用户界面,所述第一用户界面包括第一选项,所述第一选项为用于安装所述第一应用的选项;

所述用户设备检测到用户针对所述第一选项的操作。

17. 一种电子设备,其特征在于,所述电子设备包括一个或多个处理器和一个或多个存储器;其中,所述一个或多个存储器与所述一个或多个处理器耦合,所述一个或多个存储器用于存储计算机程序代码,所述计算机程序代码包括计算机指令,当所述一个或多个处理器执行所述计算机指令时,使得所述电子设备执行如权利要求4-7或8-13或14-16中任一项所述的方法。

18. 一种计算机存储介质,其特征在于,所述计算机存储介质存储有计算机程序,所述计算机程序包括程序指令,当所述程序指令在应用分发平台上运行时,使得所述应用分发平台执行如权利要求4-7或8-13或14-16中任一项所述的方法。

## 兼容性管控方法及相关设备

### 技术领域

[0001] 本申请涉及终端技术领域,尤其涉及一种兼容性管控方法及相关设备。

### 背景技术

[0002] 在当今万物互联的时代,智能终端设备的发展已经进入快车道,设备的类型变得越来越丰富,为了能够使得一套操作系统(OperatingSystem,OS)能够支持所有类型的设备,就必然要求OS有更好的可装配性。

[0003] 提高OS可装配性的一种方法是统一配置OS,然而目前不同类型的设备通常对应不同的OS,无法实现OS的统一配置,应用开发者在开发应用时也需要根据不同的OS进行差异化应用开发,灵活性差。

### 发明内容

[0004] 本申请实施例提供了一种兼容性管控方法及相关设备,可以解决设备与应用之间的兼容性问题。

[0005] 第一方面,本申请实施例提供了一种通信系统,该通信系统包括开发平台、应用分发平台、用户设备,其中:开发平台用于:获取一个或多个部件,并基于一个或多个部件生成第一标识;其中,第一标识用于指示第一设备系统能力SysCap集合,第一设备SysCap集合是用户设备的操作系统的SysCap集合;获取第二标识;其中,第二标识用于指示第一应用SysCap集合,第一应用SysCap集合是第一应用在用户设备上被安装并运行时所需要的SysCap集合;应用分发平台用于:接收开发平台发送的第二标识;获取第一标识;在基于第一标识和第二标识确定第一设备SysCap集合包括第一应用SysCap集合的情况下,向用户设备发送第一请求,第一请求用于指示用户设备安装第一应用;用户设备用于:安装第一应用。

[0006] 本申请实施例通过提供一种兼容性管控方法,在实现了使用统一的OS来支持所有类型的设备这一目标的情况下,在OS弹性部署到设备后,可以基于产品兼容性标识PCID和要求产品兼容性标识RPCID来判断应用是否可以在设备上安装并运行,若是,则可以向该设备分发该应用,从而可以解决设备与应用之间的兼容性问题。

[0007] 在一种可能的实现方式中,第一标识为第一产品兼容性标识PCID,第二标识为第一要求的产品兼容性标识RPCID。

[0008] 这样,应用分发平台可以基于第一PCID和第一RPCID来判断第一设备SysCap集合是否包括第一应用SysCap集合,若是,则向用户设备发送第一请求,若否,则不向用户设备发送第一请求。

[0009] 在一种可能的实现方式中,该通信系统还包括设备认证平台,设备认证平台用于:接收并保存开发平台发送的第一标识;向应用分发平台发送第一标识。

[0010] 这样,设备认证平台可以基于第一标识进行设备认证;还可以向应用分发平台发送第一标识,以便应用分发平台基于第一标识来生成第一设备SysCap集合。

[0011] 第二方面,本申请实施例提供了一种兼容性管控方法,应用于应用分发平台,该方法包括:应用分发平台获取第一标识,第一标识用于指示第一设备系统能力SysCap集合,第一设备SysCap集合是用户设备的操作系统的SysCap集合;应用分发平台获取第二标识,第二标识用于指示第一应用SysCap集合,第一应用SysCap集合是第一应用在用户设备上被安装并运行时所需要的SysCap集合;应用分发平台在基于第一标识和第二标识确定第一设备SysCap集合包括第一应用SysCap集合的情况下,向用户设备发送第一请求,第一请求用于指示用户设备安装第一应用。

[0012] 本申请实施例通过提供一种兼容性管控方法,应用分发平台可以基于产品兼容性标识PCID和要求产品兼容性标识RPCID来判断应用是否可以在设备上安装并运行,若是,则可以向该设备分发该应用,从而可以解决设备与应用之间的兼容性问题。

[0013] 在一种可能的实现方式中,第一标识为第一产品兼容性标识PCID,第二标识为第一要求的产品兼容性标识RPCID。

[0014] 这样,应用分发平台可以基于第一PCID和第一RPCID来判断第一设备SysCap集合是否包括第一应用SysCap集合,若是,则向用户设备发送第一请求,若否,则不向用户设备发送第一请求。

[0015] 在一种可能的实现方式中,应用分发平台获取第一标识,具体包括:应用分发平台向设备认证平台获取第一PCID;其中,第一PCID是开发平台基于第一设备SysCap集合生成并发送给设备认证平台的。

[0016] 这样,应用分发平台可以基于第一PCID来生成第一设备SysCap集合。

[0017] 在一种可能的实现方式中,应用分发平台获取第二标识,具体包括:应用分发平台接收开发平台发送的第一RPCID;其中,第一RPCID是开发平台基于第一应用SysCap集合生成的,第一应用SysCap集合是应用分发平台基于软件开发工具包SDK发布平台上发布的第一SDK或第一PCID生成的。

[0018] 这样,应用分发平台可以基于第一RPCID来生成第一应用SysCap集合。

[0019] 第三方面,本申请实施例提供了一种兼容性管控方法,应用于开发平台,其特征在于,该方法包括:开发平台获取第一信息,第一信息包括第一软件开发工具包SDK或第一标识,第一标识用于指示第一设备系统能力SysCap集合,第一设备SysCap集合是用户设备的操作系统的SysCap集合;开发平台基于第一信息生成第二标识,第二标识用于指示第一应用SysCap集合,第一应用SysCap集合是第一应用在用户设备上被安装并运行时所需要的SysCap集合;开发平台向应用分发平台发送第二标识。

[0020] 在一种可能的实现方式中,第一标识为第一产品兼容性标识PCID,第二标识为第一要求的产品兼容性标识RPCID。

[0021] 在一种可能的实现方式中,开发平台基于第一信息生成第二标识,具体包括:在用户设备的设备类型为第一类型的情况下,开发平台基于第一SDK生成第一RPCID;或,在用户设备的设备类型不为第一类型的情况下,开发平台基于第一PCID生成第一RPCID。

[0022] 这样,在用户设备为第一类型的设备(例如手机、智慧屏、PC、平板等一些爆款电子设备)的情况下,开发平台可以基于SDK来生成RPCID,在用户设备不为第一类型的设备的情况下,开发平台可以基于PCID来生成RPCID。

[0023] 在一种可能的实现方式中,在开发平台获取第一信息之前,该方法还包括:开发平

台获取一个或多个部件,并基于一个或多个部件生成第一标识。

[0024] 这样,开发平台可以从操作系统开发平台获取一个或多个部件来进行拼装,组成一个操作系统,基于这一个或多个部件来生成设备SysCap集合,并对设备SysCap集合进行编码生成PCID。

[0025] 在一种可能的实现方式中,开发平台基于一个或多个部件生成第一标识,具体包括:开发平台基于一个或多个部件生成第一设备SysCap集合;开发平台对第一设备SysCap集合进行编码生成第一PCID。

[0026] 在一种可能的实现方式中,在开发平台基于一个或多个部件生成第一标识之后,该方法还包括:开发平台向设备认证平台发送第二请求,第二请求中包括第一PCID;在设备认证平台基于第一PCID确认设备认证通过的情况下,开发平台接收并保存设备认证平台发送的第一令牌Token。

[0027] 这样,开发平台可以向设备认证平台发送设备认证请求,在设备认证通过后,可以接收到设备认证平台发送的Token,提高安全性。

[0028] 第四方面,本申请实施例提供了一种兼容性管控方法,应用于用户设备,该方法包括:用户设备接收应用分发平台发送的第一请求,第一请求中包括第一要求的产品兼容性标识RPCID;在用户设备基于第一RPCID和第一产品兼容性标识PCID确定第一设备SysCap集合包括第一应用SysCap集合的情况下,用户设备安装第一应用;其中,第一RPCID用于指示第一应用SysCap集合,第一应用SysCap集合是第一应用在用户设备上被安装并运行时所需要的SysCap集合,第一PCID用于指示第一设备系统能力SysCap集合,第一设备SysCap集合是用户设备的操作系统的SysCap集合。

[0029] 在一种可能的实现方式中,用户设备包括应用包管理服务PMS和Samgr部件,用户设备基于第一RPCID和第一产品兼容性标识PCID确定第一设备SysCap集合包括第一应用SysCap集合,具体包括:用户设备利用PMS对第一RPCID进行解码生成第一应用SysCap集合;用户设备利用Samgr部件查询第一设备SysCap集合;用户设备利用PMS确定第一设备SysCap集合包括第一应用SysCap集合。

[0030] 这样,用户设备在安装并运行应用之前可以先判断第一设备SysCap集合是否包括第一应用SysCap集合,若是,再安装该应用。

[0031] 在一种可能的实现方式中,在用户设备安装第一应用之前,该方法还包括:用户设备显示第一用户界面,第一用户界面包括第一选项,第一选项为用于安装第一应用的选项;用户设备检测到用户针对第一选项的操作。

[0032] 这样,用户设备可以显示可以安装的应用,在用户同意安装的情况下,用户设备再安装该应用。

[0033] 第五方面,本申请实施例提供了一种兼容性管控方法,应用于设备认证平台,该方法包括:设备认证平台接收开发平台发送的第二请求,第二请求中包括第一产品兼容性标识PCID;设备认证平台基于第一PCID确认设备认证通过,生成第一令牌Token,并保存第一PCID与第一令牌Token的映射关系;设备认证平台向开发平台发送第一令牌Token。

[0034] 在一种可能的实现方式中,在设备认证平台向开发平台发送第一令牌Token之后,该方法还包括:设备认证平台接收应用分发平台发送的第三请求;设备认证平台基于第三请求向应用分发平台发送第一PCID。

[0035] 第六方面,本申请实施例提供了一种电子设备,该电子设备包括一个或多个处理器和一个或多个存储器;其中,一个或多个存储器与一个或多个处理器耦合,一个或多个存储器用于存储计算机程序代码,计算机程序代码包括计算机指令,当一个或多个处理器执行计算机指令时,使得电子设备执行上述第二方面或第三方面或第四方面或第五方面任一项可能的实现方式中的方法。

[0036] 第七方面,本申请实施例提供了一种计算机存储介质,该计算机存储介质存储有计算机程序,计算机程序包括程序指令,当程序指令在电子设备上运行时,使得电子设备执行上述第二方面或第三方面或第四方面或第五方面任一项可能的实现方式中的方法。

[0037] 第八方面,本申请实施例提供了一种计算机程序产品,当计算机程序产品在计算机上运行时,使得计算机执行上述第二方面或第三方面或第四方面或第五方面任一项可能的实现方式中的方法。

## 附图说明

[0038] 图1是本申请实施例提供的一种软件架构示意图;

[0039] 图2是本申请实施例提供的一种通信系统的架构示意图;

[0040] 图3是本申请实施例提供的一种基于操作系统源码生成的归一化的SDK示意图;

[0041] 图4是本申请实施例提供的一种产品兼容性标识PCID的结构示意图;

[0042] 图5A-图5C是本申请实施例提供的一组用户设备中包括的部件集合示意图;

[0043] 图6是本申请实施例提供的一种要求产品兼容性标识RPCID的结构示意图;

[0044] 图7是本申请实施例提供的一种兼容性管控方法的流程示意图;

[0045] 图8是本申请实施例提供的一种应用开发平台生成应用的RPCID的流程示意图;

[0046] 图9是本申请实施例提供的一种用户设备进行应用安装的流程示意图;

[0047] 图10是本申请实施例提供的一种通过用户设备中的Samgr 部件进行SysCap注册与查询过程的示意图;

[0048] 图11A-图11D是本申请实施例提供的一组用户界面示意图;

[0049] 图12是本申请实施例提供的一种用户设备的软件结构示意图;

[0050] 图13是本申请实施例提供的一种用户设备的结构示意图。

## 具体实施方式

[0051] 下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述。其中,在本申请实施例的描述中,除非另有说明,“/”表示或的意思,例如,A/B可以表示A或B;文本中的“和/或”仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如,A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况,另外,在本申请实施例的描述中,“多个”是指两个或两个以上。

[0052] 应当理解,本申请的说明书和权利要求书及附图中的术语“第一”、“第二”等是用于区别不同对象,而不是用于描述特定顺序。此外,术语“包括”和“具有”以及它们任何变形,意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有限定于已列出的步骤或单元,而是可选地还包括没有列出的步骤或单元,或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或单元。

[0053] 在本申请中提及“实施例”意味着,结合实施例描述的特定特征、结构或特性可以包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例,也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是,本申请所描述的实施例可以与其它实施例相结合。

[0054] 为了便于理解,下面对本申请实施例中涉及的一些相关概念进行说明。

[0055] 1、产品兼容性标识(ProductCompatibilityID,PCID)

[0056] PCID,也可以称为设备兼容性标识或者设备系统能力标识,用于标记设备支持的基础系统类型(SystemType)和选配的系统能力(SystemCapability, SysCap)。

[0057] 其中,系统类型可以包括但不限于轻量系统(例如智能家居领域的连接类模组、传感器设备、穿戴类设备等)、小型系统(例如智能家居领域的路由器、电子猫眼、行车记录仪等)、标准系统(例如带屏的物联网设备等)、大型系统(例如智慧屏、智能手机等)。

[0058] 其中,系统能力可以是指设备的操作系统所具备的特性或功能,设备的系统能力可以由组成设备的操作系统的部件集合来提供的。

[0059] 在本申请实施例中,PCID可以是通过对API版本信息(APIVersion)、系统类型、设备厂商ID(ManufactureID,MID)、以及组成设备的操作系统的部件集合进行编码生成的。具体可以参照后续实施例中的相关内容,在此先不展开。

[0060] 2、要求的产品兼容性标识(RequiredProductCompatibilityID,RPCID)

[0061] RPCID,也可以称为应用兼容性标识,用于标记应用静态依赖的系统能力,即应用要求的系统能力集合。

[0062] 在本申请实施例中,RPCID可以是通过对API版本信息(APIVersion)、应用要求的系统能力集合进行编码生成的,其中,应用要求的系统能力集合可以是指设备的操作系统安装并运行该应用所需要具备的部件集合。具体可以参照后续实施例中的相关内容,在此先不展开。

[0063] 3、南向接口、北向接口

[0064] 南向接口(Southbound Interface)是指向下的接口,即高层次连接低层次的接口。在本申请实施例中,南向接口可以是设备提供的接口,也即是说,南向可以是针对设备而言。

[0065] 北向接口(Northbound Interface)是指向上的接口,即低层次连接高层次的接口。在本申请实施例中,北向接口可以是应用提供的接口,也即是说,北向可以是针对应用而言。

[0066] 操作系统(OperatingSystem,OS)南北向生态兼容性直接决定了一个操作系统生态的健康程度,解决好这个生态兼容性问题是一个操作系统成功的基础。

[0067] 目前,设备厂商(即南向开发者)通常将不同设备类型(Device Type)的设备对应不同的OS,应用开发者(即北向开发者)在开发应用时根据不同的OS进行差异化应用开发,从而保证南北向的生态兼容性。

[0068] 以Android的操作系统为例,Android为不同设备类型的设备开发了不同的OS,例如,开发了为Phone/Pad使用的Android;开发了为TV使用的AndroidTV;开发了为Wear设备使用的AndroidWear;开发了为IoT设备使用的Android Things;开发了为PC使用的Chrome OS for PC;开发了为汽车使用的Android Automotive;等等。为了支持特定设备类型的使

用场景,通常是在Android公共平台上提供一些额外的接口,如图1所示,Android Automotive是在Android公共平台上叠加了汽车应用程序编程接口(Car Application Programming Interface,Car API)。

[0069] 与Android类似,Apple也是为不同设备类型的设备开发了不同的OS,例如,开发了为Phone使用的IOS;开发了为Pad使用的Ipad OS;开发了为Wear设备使用的Watch OS;开发了为PC使用的Mac OS;等等。

[0070] 随着终端技术的不断发展,终端设备的设备类型越来越多,需要不断地新增设备类型定义,如果针对每个设备类型的设备都开发一套OS,就容易导致基于设备类型进行应用分发的可扩展性不足问题,此外,针对相同设备类型的设备,如果存在设备能力上的差异,应用开发者在开发应用时还需要进行应用差异化分发,灵活性较差。

[0071] 因此,如何提高OS的可装配性,使用统一的OS来支持所有设备类型的设备(即统一配置OS),并且能够保证操作系统南北向生态的兼容性,是亟待解决的问题。

[0072] 本申请实施例提供了一种基于PCID和RPCID的兼容性管控方法,在实现了使用统一的OS来支持所有类型的设备这一目标的情况下,在OS弹性部署到设备后,可以解决设备与应用之间的兼容性问题,即可以保证南北向生态的兼容性。

[0073] 后续实施例会详细介绍具体的实现方法,在此先不赘述。

[0074] 下面介绍本申请实施例提供的一种通信系统。

[0075] 图2示例性示出了本申请实施例提供的一种通信系统。

[0076] 如图2所示,该通信系统可以包括操作系统开发平台201、软件开发工具包(SoftwareDevelopmentKit,SDK)发布平台202、设备开发平台203、设备认证平台204、应用开发平台205、应用分发平台206、用户设备200。其中:

[0077] 操作系统开发平台201,可以用于开发操作系统,提供操作系统源码。

[0078] 操作系统源码可以包括必选部件集合和可选部件集合,部件可提供的系统能力可以定义为SysCap。

[0079] 其中,必选部件集合可以包括一个或多个必选部件,例如部件A(也可以称为SysCapA)等等;可选部件集合可以包括一个或多个可选部件,例如部件B(也可以称为SysCapB)等等。

[0080] 在本申请实施例中,操作系统源码可以是指OpenHarmony提供的鸿蒙操作系统(HarmonyOS)源码。

[0081] 软件开发工具包发布平台202(也可以称为SDK子系统),可以用于基于一套操作系统源码生成归一化的SDK。

[0082] 在本申请实施例中,软件开发工具包发布平台202可以基于操作系统开发平台201提供的操作系统源码来生成并发布归一化的SDK,从而满足多设备的应用开发。

[0083] 如图3所示,归一化的SDK是基于操作系统开发平台201提供的操作系统源码生成的。在本申请实施例中,软件开发工具包发布平台202生成的归一化的SDK除了可以提供编译工具链、运行时(Run Time)、开发工具、应用编程接口

(ApplicationProgrammingInterface,API)文档之外,还可以新增如下两个功能:

[0084] 1、可以在各个部件的API中增加SysCap属性,在SDK生成时为API添加该属性,这样,可以实现API与系统能力之间的关联,在应用开发时可以根据其调用的API对应的

SysCap属性来要求该应用运行时依赖的系统能力。

[0085] 继续参阅图3,在API中增加SysCap属性具体可以是在API中标注“@SysCap”。例如,API-1与部件A相关联,则可以在API-1中标注“@SysCapA”,即API-1@SysCapA;又例如API-2与部件C相关联,则可以在API-2中标注“@SysCapC”,即API-2@SysCapC;再例如API-3与部件B相关联,则可以在API-3中标注“@SysCapB”,即API-3@SysCapB;等等。

[0086] 2、可以定义爆款电子设备对应设备类型的SysCap集合。

[0087] 继续参阅图3,其中,爆款电子设备对应设备类型可以包括但不限于以下几种:手机、PC、儿童表、TV、平板。

[0088] 示例性地,如图3所示,手机对应的SysCap集合可以为{SysCapA、SysCapB、…、SysCapX};PC对应的SysCap集合可以为{SysCapA、SysCapD、…、SysCapY};儿童表对应的SysCap集合可以为{SysCapC、SysCapE、…、SysCapZ};TV对应的SysCap集合可以为{SysCapC、SysCapE、…、SysCapM};平板对应的SysCap集合可以为{SysCapA、SysCapE、…、SysCapN};等等。

[0089] 设备开发平台203,可以用于支持设备厂商在操作系统开发平台201上获取操作系统部件,并按需拼装操作系统部件,得到系统部件集合(也可以称为OS部件集合)。

[0090] 还可以用于支持设备厂商定义私有部件集合,并将私有部件集合与OS部件集合组合成完整的设备部件集合(也可以称为设备SysCap集合),生成一套操作系统。

[0091] 还可以用于支持设备厂商将生成的操作系统部署到用户设备200。

[0092] 还可以用于将设备SysCap集合转化为设备的产品兼容性标识PCID,并向设备认证平台204发送PCID,以使得设备认证平台204基于PCID进行设备认证。

[0093] 下面基于图4介绍PCID的构成:

[0094] 如图4所示,PCID可以由以下几个字段构成:API版本信息(APIVersion)、系统类型、设备厂商ID(ManufactureID,MID)、系统SysCap集合、私有SysCap集合。其中:

[0095] API版本信息,可以包括但不限于版本名称、版本号、生成时间等信息,bit0-bit15可以用于表示API版本信息,其中,PCID中的API版本信息可以取设备开发时采用的操作系统的API版本信息,bit15固定设置为0。

[0096] 系统类型,可以包括但不限于轻型系统或称轻量系统(Mini System)、小型系统(SmallSystem)、标准系统(StandardSystem)、大型系统(LargeSystem)等,bit16-bit31可以用于表示系统类型,其中,bit16可以表示轻型系统;bit17可以表示小型系统;bit18可以表示标准系统;bit19可以表示大型系统;bit20-bit31可以表示未定义(reserved)的系统。在PCID中,bit16-bit31中某一个bit设置为1,则系统类型为该bit所表示的系统类型(例如bit16为1,则表示系统类型为轻型系统),如果默认所有bit全为0,则表示系统类型未定义。

[0097] 设备厂商ID,用于标识一个唯一的设备厂商信息。bit32-bit63可以用于表示设备厂商ID。如果不存在扩展的系统能力,则此字段可以设置为0。

[0098] 系统SysCap集合,该字段采用TLV格式封装,其中,T为Type(类型),共32bits,bit31固定为0,表示该SysCap集合是系统定义的SysCap集合;L为Length(长度),共16bits,表示Value字段的总长度;V为Value(值),大于或等于16bits,表示系统定义的SysCapID。

[0099] 私有SysCap集合,该字段同样采用TLV格式封装,其中,T为Type,共32bits,bit31固定为1,表示该SysCap集合是设备厂商扩展的SysCap集合;L为Length,共16bits,表示

Value字段的总长度;V为Value,大于或等于16bits,表示设备厂商扩展的SysCapID。

[0100] 在本申请实施例中,系统能力可以区分不同的属性组,例如,基本部件集合(BaseComponentGroup,BCG)、可选部件集合(OptionalComponentGroup,OCG)、私有部件集合(PrivateComponentGroup,PCG),在面向多设备部署时,支持采用“BCG+OCG+PCG”方式定义特定设备的系统能力;其中,BCG可以是某一个系统类型对应的最小部件集合(例如操作系统开发平台201中提供的必选部件集合),OCG可以是系统定义的可选部件集合(例如操作系统开发平台201中提供的可选部件集合),PCG可以是设备厂商定义的私有部件集合。

[0101] 在本申请实施例中,“系统类型”这一字段可以用于表示该系统类型对应的BCG(例如图5A示例性所示的鸿蒙认证最小部件集合);“系统SysCap集合”这一字段可以用于表示OCG(例如图5B示例性所示的鸿蒙可选部件集合);“私有SysCap集合”这一字段可以用于表示PCG(例如图5C示例性所示的Vendor自研部件集合)。

[0102] 设备认证平台204,也可以称为设备认证中心或者云端认证中心,可以用于进行设备认证。

[0103] 具体地,设备认证平台204可以接收设备开发平台203发送的PCID,之后可以基于该PCID中包含的SysCap集合进行认证,认证通过之后可以生成设备令牌(Token),并将该PCID与设备Token进行映射,保存PCID与设备Token之间的映射关系,之后可以将该PCID对应的设备Token发送给设备开发平台203。

[0104] 容易理解,设备认证平台204可以保存如下表1所示的一个或多个PCID与设备Token之间的映射关系。

PCID	设备Token
PCID1	Token1
PCID2	Token2
...	...
PCIDx	Tokenx

[0105] 表1

[0106] 设备认证平台204还可以用于提供查询和下载PCID的功能。

[0107] 例如,应用分发平台206可以向设备认证平台204获取PCID。一种可能的实现方式是,应用分发平台206可以向设备认证平台204发送设备Token,设备认证平台204可以基于已保存的PCID与设备Token之间的映射关系查询与该设备Token相对应的PCID,并将该PCID发送给应用分发平台206。

[0108] 应用开发平台205,可以用于进行跨设备的应用开发。

[0109] 示例性地,应用开发平台205可以向软件开发工具包发布平台202获取SDK,之后可以利用集成开发环境(IntegratedDevelopmentEnvironment,IDE)基于获取到的SDK来进行应用开发,并定义应用的要求产品兼容性标识RPCID。进一步地,在已开发的应用需要上架时,可以向应用分发平台206发送该应用的RPCID。

[0110] 下面基于图6介绍RPCID的构成:

[0111] 如图6所示,RPCID可以由以下两个字段构成:API版本信息(APIVersion)、要求的SysCap集合。其中:

[0112] API版本信息,可以包括但不限于版本名称、版本号、生成时间等信息,bit0-bit15

可以用于表示API版本信息,其中,RPCID中的API版本信息可以取应用开发时采用的SDK中的API版本信息,bit15固定设置为1。

[0114] 要求的SysCap集合,该字段采用TLV格式封装,其中,T为Type(类型),共32bits,用于表示该应用适配的设备类型(设备类型与特定SysCap集合存在一一映射关系);L为Length(长度),共16bits,表示Value字段的总长度;V为Value(值),大于或等于16bits,表示该应用所需的SysCapID。

[0115] 应用分发平台206,也可以称为应用市场服务器或应用商店服务器,可以用于接收并保存应用开发平台205发送的应用的RPCID;还可以用于将应用的RPCID与设备的PCID进行匹配,在确认应用的RPCID的范围小于或等于设备的PCID的范围的情况下,向该PCID相对应的设备(例如用户设备200)分发应用。其中,设备的PCID是应用分发平台206向设备认证平台204获取的。

[0116] 对于同一个部件的组件/控件在不同的设备上的差异,通过应用配置中的distributefilter字段进行分发。

[0117] 用户设备200,可以用于进行应用安装,并运行已安装的应用。

[0118] 具体地,用户设备200可以包括包管理服务模块(PackageManagerService,PMS),PMS可以用于解析应用的RPCID与设备的PCID,并判断应用所需的SysCap集合是否小于或等于设备具备的SysCap集合,若是,则PMS可以确认该应用可以在用户设备200上正常安装并运行。

[0119] 用户设备200还可以包括系统服务管理模块(也可以称为系统服务管理部件),其中,系统管理服务模块中可以包括Samgr部件,Samgr部件是OpenHarmony的核心部件,可以用于提供OpenHarmony系统服务启动、注册、查询等功能。

[0120] 在本申请实施例中,系统管理服务模块(例如Samgr部件)可以为系统各个部件提供SysCap的注册和查询接口,还可以为应用提供SysCap的查询接口,应用可以在调用某个API之前检测系统是否具备与该API相关联的SysCap。

[0121] 在一些实施例中,设备开发平台203和应用开发平台205可以为同一个平台,例如,可以统称为开发平台。

[0122] 下面介绍本申请实施例提供的一种兼容性管控方法。

[0123] 图7示例性示出了本申请实施例提供的一种兼容性管控方法的具体流程。

[0124] 如图7所示,该方法可以应用于包括设备开发平台、设备认证平台、应用开发平台、应用分发平台、用户设备的通信系统。下面详细介绍该方法的具体步骤:

[0125] 阶段一:设备开发阶段

[0126] S701、设备开发平台拼装OS部件,生成设备SysCap集合1,并基于设备SysCap集合1编译生成设备SysCap配置文件1。

[0127] 具体地,设备开发平台可以在操作系统开发平台上获取OS部件,设备厂商可以在设备开发平台上按需拼装OS部件,生成设备SysCap集合1。进一步地,设备开发平台可以基于设备SysCap集合1编译生成设备SysCap配置文件1。

[0128] 其中,设备SysCap配置文件1中可以包括但不限于API版本信息、设备厂商ID、设备SysCap集合1(也可以称为设备SysCap列表1)。

[0129] 在一些实施例中,设备SysCap集合1中不仅可以包括OS部件集合,还可以包括设备

厂商定义的私有部件集合。

[0130] S702、设备开发平台利用SysCap编解码工具对设备SysCap配置文件1进行编码生成PCID1。

[0131] 具体地,在生成设备SysCap配置文件1之后,设备开发平台可以将设备SysCap配置文件1中的API版本信息、设备厂商ID、设备SysCap集合1等信息作为SysCap编解码工具的输入数据,利用SysCap编解码工具对设备SysCap配置文件1进行编码生成PCID1。

[0132] 需要说明的是,本申请实施例对SysCap编解码工具对设备SysCap配置文件1进行编码时使用的编码方式不作限定。

[0133] 阶段二:设备认证阶段

[0134] S703、设备开发平台向设备认证平台发送设备认证请求1,该请求中包括PCID1。

[0135] 具体地,设备开发平台在生成PCID1之后,可以向设备认证平台发送设备认证请求1,该设备认证请求1中可以包括PCID1。

[0136] 其中,PCID1可以用于设备认证平台基于该PCID1来进行设备认证。

[0137] S704-S706、设备认证平台确认本次设备认证通过,生成设备令牌Token1,并保存PCID1与Token1的映射关系。之后,设备认证平台可以向设备开发平台发送Token1,设备开发平台接收到设备认证平台发送的Token1之后,可以保存Token1。

[0138] 具体地,设备认证平台在接收到设备开发平台发送的PCID1之后,可以基于该PCID1中包含的SysCap集合进行设备认证,在确认本次设备认证通过之后,可以生成设备令牌Token1,并可以保存PCID1、保存PCID1与Token1的映射关系。

[0139] 之后,设备认证平台可以向设备开发平台发送Token1,设备开发平台在接收到设备认证平台发送的Token1之后,可以保存Token1。

[0140] 其中,Token1可以用于在后续步骤中向设备认证平台获取PCID1。

[0141] 阶段三:应用开发阶段

[0142] S707、应用开发平台基于SDK1或PCID1生成设备SysCap集合1,并进行API静态检查。

[0143] 对于不同的设备类型,应用开发平台生成设备SysCap集合的方式也不同。

[0144] 在本申请实施例中,设备类型可以分为两大类:1+8设备、N设备。

[0145] 其中,1+8设备可以为爆款电子设备,爆款电子设备可以包括手机、PC、平板、智慧屏、音箱、手表、眼镜、车机、耳机。

[0146] 其中,N设备可以为除了上述爆款电子设备之外的其他电子设备,例如,摄像头、扫地机器人、智能秤等涵盖移动办公、智能家居、运动健康、影音娱乐、智慧出行等多种场景模式的电子设备。

[0147] 参阅图8,首先,在开发应用之前,应用开发平台可以先判断适配该应用的设备类型(即步骤S801)。

[0148] 若应用开发平台确认设备类型为N设备(即步骤S802a),则应用开发平台可以利用SysCap编解码工具对应用开发者导入的设备的PCID进行解码生成设备SysCap集合(即步骤S803a)。

[0149] 若应用开发平台确认设备类型为1+8设备(即步骤S802b),由于SDK发布平台上已经发布了1+8设备对应的SDK,且SDK中已经定义了1+8设备对应的设备SysCap集合,则应用

开发平台可以向SDK发布平台获取1+8设备对应的SDK,并基于SDK获取设备SysCap集合(即步骤S803b)。

[0150] 进一步地,在生成设备SysCap集合之后,应用开发平台可以基于设备SysCap集合以及API的SysCap属性进行API静态检查(即步骤S805),提升开发者体验。可选地,还可以进行API联想。

[0151] 其中,API静态检查(即对API的调用做静态检查)和API联想均可以使用现有技术,在此不再赘述。

[0152] 容易理解,如果SDK1为1+8设备(例如手机)对应的SDK,则应用开发平台可以基于SDK1生成设备SysCap集合1;如果PCID1为N设备(例如扫地机器人)对应的PCID,则应用开发平台可以基于PCID1生成设备SysCap集合1。

[0153] S708、应用开发平台基于设备SysCap集合1和应用SysCap集合1编译生成应用SysCap配置文件1。

[0154] 继续参阅图8,应用开发平台可以基于设备SysCap集合和应用SysCap集合编译生成应用SysCap配置文件(即步骤S804),其中,应用SysCap配置文件中可以包括“设备类型(device\_type)”字段和“需要的设备SysCap(required\_device\_syscap)”字段。

[0155] 不同的设备类型,应用SysCap配置文件中“设备类型(device\_type)”字段和“需要的设备SysCap(required\_device\_syscap)”字段的设置方式不同。

[0156] 假设设备SysCap集合为设备SysCap集合1,应用SysCap集合为应用SysCap集合1,应用SysCap配置文件为应用SysCap配置文件1。

[0157] 如果设备SysCap集合1是通过解码设备的PCID得到的,则适配该应用的设备类型为N设备,进一步地,应用开发平台可以对“设备类型(device\_type)”字段对应的内容设置为空,应用开发平台可以在“需要的设备SysCap(required\_device\_syscap)”字段对该应用所需的SysCap进行全量定义,即将该应用所需的SysCap全部写入应用SysCap配置文件1中,例如,该应用所需的SysCap为SysCapA、SysCapB、SysCapC,则应用配置文件1中“需要的设备SysCap(required\_device\_syscap)”字段对应的内容为SysCapA、SysCapB、SysCapC。容易理解,在这种情况下,应用SysCap集合1中只包括应用配置文件1中“需要的设备SysCap(required\_device\_syscap)”字段对应的SysCap,例如SysCapA、SysCapB、SysCapC。

[0158] 如果设备SysCap集合1是通过SDK获取到的,则适配该应用的设备类型为1+8设备,进一步地,应用开发平台可以对“设备类型(device\_type)”字段对应的内容不设置为空,如果应用所需的SysCap包括对应设备类型所包括的SysCap集合之外的SysCap,则应用开发平台可以在“需要的设备SysCap(required\_device\_syscap)”字段对该应用所需的SysCap包括对应设备类型所包括的SysCap集合之外的SysCap进行增量定义,即将该应用所需的SysCap包括对应设备类型所包括的SysCap集合之外的SysCap以增量的形式写入应用SysCap配置文件1中。举例来说,设备类型为手机,则应用SysCap配置文件1中“设备类型(device\_type)”字段对应的内容为phone,如果该应用所需的SysCap包括手机这一设备类型所包括的SysCap集合之外的SysCap,例如SysCapD、SysCapE,则应用配置文件1中“需要的设备SysCap(required\_device\_syscap)”字段对应的内容为SysCapD、SysCapE。容易理解,在这种情况下,应用SysCap集合1中不仅包括应用配置文件1中“需要的设备SysCap(required\_device\_syscap)”字段对应的SysCap,例如SysCapD、SysCapE,还包括设备

SysCap集合1中包括的SysCap。

[0159] S709、应用开发平台利用SysCap编解码工具对应用SysCap配置文件1进行编码生成RPCID1。

[0160] 继续参阅图8,在生成应用SysCap配置文件之后,应用开发平台可以利用SysCap编解码工具对应用SysCap配置文件进行编码生成应用的RPCID(即步骤S806)。

[0161] 其中,应用的RPCID中的“API版本信息”这一字段的内容可以是基于SDK或PCID中的API版本信息得到的。

[0162] 容易理解,在设备类型为1+8设备的情况下,应用的RPCID中的“API版本信息”这一字段的内容可以是基于SDK中的API版本信息得到的;在设备类型为N设备的情况下,应用的RPCID中的“API版本信息”这一字段的内容可以是基于PCID中的API版本信息得到的。

[0163] 其中,应用的RPCID中的“要求的SysCap集合”这一字段的内容可以是只基于应用SysCap配置文件中“需要的设备SysCap(required\_device\_syscap)”字段对应的SysCap得到的;或,也可以是基于应用SysCap配置文件中“需要的设备SysCap(required\_device\_syscap)”字段对应的SysCap以及“设备类型(device\_type)”字段对应的设备类型所包括的SysCap得到的。

[0164] 容易理解,在设备类型为1+8设备的情况下,应用的RPCID中的“要求的SysCap集合”这一字段的内容可以是基于应用SysCap配置文件中“需要的设备SysCap(required\_device\_syscap)”字段对应的SysCap以及“设备类型(device\_type)”字段对应的设备类型所包括的SysCap得到的;在设备类型为N设备的情况下,应用的RPCID中的“要求的SysCap集合”这一字段的内容可以是只基于应用SysCap配置文件中“需要的设备SysCap(required\_device\_syscap)”字段对应的SysCap得到的。

[0165] 在本申请实施例中,应用开发平台可以利用SysCap编解码工具对应用SysCap配置文件1进行编码生成应用的RPCID1。

[0166] S710-S711、应用开发平台向应用分发平台发送RPCID1,应用分发平台接收到应用开发平台发送的RPCID1之后,可以保存RPCID1。

[0167] 具体地,应用开发平台在生成RPCID1之后,在RPCID1对应的应用需要上架时,应用开发平台可以向应用分发平台发送RPCID1,应用分发平台在接收到应用开发平台发送的RPCID1之后,可以保存RPCID1,以便后续应用分发平台可以基于RPCID1生成应用对应的SysCap集合(即应用SysCap集合)。

[0168] 阶段四:应用分发阶段

[0169] S712-S714、设备开发平台向用户设备进行OS部署(发送设备SysCap集合1和Token1等),用户设备保存设备SysCap集合1和Token1,并向应用分发平台发送Token1。

[0170] 具体地,设备开发平台在通过设备认证之后(即步骤S706执行完成之后),可以向用户设备进行OS部署。示例性地,设备开发平台可以向用户设备发送设备SysCap集合1和Token1等信息,用户设备接收到设备开发平台发送的设备SysCap集合1和Token1等信息之后,可以保存设备SysCap集合1和Token1等信息,进一步地,用户设备可以基于设备SysCap集合1部署OS。

[0171] 进一步地,用户设备还可以向应用分发平台发送Token1,以便应用分发平台可以基于Token1向设备认证平台获取Token1对应的PCID。

[0172] 需要说明的是,本申请实施例对执行步骤S712-步骤S714的时机不作限定。例如,步骤S712-步骤S714可以是在步骤S706执行完就开始执行;又例如,步骤S712-步骤S714也可以是与阶段三中的步骤同时执行;再例如,步骤S712-步骤S714也可以是在阶段三中的步骤执行完之后再执行;等等。

[0173] S715-S717、应用分发平台向设备认证平台发送获取PCID的请求,该请求中可以包括Token1,设备认证平台接收到应用分发平台发送的获取PCID的请求之后,可以基于PCID与Token的映射关系确认Token1对应PCID1。之后,设备认证平台可以向应用分发平台发送PCID1。

[0174] 具体地,在接收到用户设备发送的Token1之后,应用分发平台可以向设备认证平台发送获取PCID的请求,该请求中可以包括Token1,设备认证平台在接收到应用分发平台发送的获取PCID的请求之后,由于设备认证平台提前保存有PCID与Token的映射关系,因此,设备认证平台可以基于提前保存的PCID与Token的映射关系确认Token1对应的PCID为PCID1。之后,设备认证平台可以向应用分发平台发送PCID1。

[0175] 在一些实施例中,步骤S715是可选的。例如,设备认证平台可以主动向应用分发平台发送全部的PCID和Token以及每个PCID和Token的映射关系,应用分发平台可以基于每个PCID和Token的映射关系来确定Token1对应PCID1。

[0176] 在一些实施例中,在步骤S715为可选步骤的情况下,步骤S714也是可选的,也即是说,在无需执行步骤S715的情况下,步骤S714也无需执行。

[0177] S718、应用分发平台利用SysCap编解码工具对PCID1和RPCID1进行解码依次得到设备SysCap集合1、应用SysCap集合1。

[0178] 具体地,在接收到设备认证平台发送的PCID1之后,应用分发平台可以利用SysCap编解码工具PCID1进行解码得到设备SysCap集合1。

[0179] 应用分发平台也可以利用SysCap编解码工具对RPCID1进行解码得到应用SysCap集合1。其中,RPCID1是应用开发平台发送给应用分发平台,应用分发平台保存下来的。

[0180] 需要说明的是,本申请实施例对应用分发平台利用SysCap编解码工具对RPCID1进行解码得到应用SysCap集合1的时机不作限定。例如,应用分发平台可以在接收到应用开发平台发送的RPCID1之后(即步骤S710执行完成之后)就利用SysCap编解码工具对RPCID1进行解码得到应用SysCap集合1;又例如,应用分发平台可以在接收到设备认证平台发送的PCID1(即步骤S717执行完成之后)之后,再利用SysCap编解码工具对提前保存的RPCID1(即步骤S711中保存的RPCID1)进行解码得到应用SysCap集合1。

[0181] S719、应用分发平台确认应用SysCap集合1的范围小于或等于设备SysCap集合1的范围。

[0182] 具体地,在解码得到设备SysCap集合1、应用SysCap集合1之后,应用分发平台可以判断应用SysCap集合1的范围是否小于或等于设备SysCap集合1的范围,若是,则应用分发平台则继续执行下述步骤S720,即应用分发平台确认用户设备具备应用SysCap集合1对应的应用所必需的系统能力,并向用户设备分发应用SysCap集合1对应的应用;若否,则应用分发平台不再执行下述步骤S720,即应用分发平台确认用户设备不具备应用SysCap集合1对应的应用所必需的系统能力,不会向用户设备分发应用SysCap集合1对应的应用。

[0183] 容易理解,在本申请实施例中,应用SysCap集合1的范围小于或等于设备SysCap集

合1的范围指的是应用SysCap集合1包含于设备SysCap集合1中,即应用SysCap集合1是设备SysCap集合1的子集。

[0184] S720、应用分发平台向用户设备发送应用安装请求,该请求中包括RPCID1。

[0185] 具体地,应用分发平台在确认应用SysCap集合1的范围小于或等于设备SysCap集合1的范围之后,可以向用户设备发送应用安装请求(即安装应用SysCap集合1对应的应用的请求),该请求中包括RPCID1。

[0186] 阶段五:应用安装阶段

[0187] S721、用户设备利用SysCap编解码工具对RPCID1进行解码得到应用SysCap集合1。

[0188] 具体地,在接收到应用分发平台发送的应用安装请求之后,参阅图9,用户设备的包管理服务模块PMS可以利用SysCap编解码工具对应用安装请求中携带的RPCID进行解码得到应用SysCap集合(即步骤S901)。

[0189] 示例性地,在上述步骤S720中,应用安装请求中携带的RPCID是RPCID1,则用户设备在接收到应用分发平台发送的应用安装请求之后,用户设备的PMS可以利用SysCap编解码工具对RPCID1进行解码得到应用SysCap集合1。

[0190] S722、用户设备判断应用SysCap集合1的范围是否小于或等于当前设备SysCap集合的范围,若是,则确认该应用可以在设备上安装并正常运行。

[0191] 具体地,参阅图9,在接收到应用分发平台发送的应用安装请求之后,用户设备的PMS除了可以利用SysCap编解码工具对应用安装请求中的RPCID1进行解码得到应用SysCap集合1,还可以通过调用系统管理服务模块中Samgr部件提供的SysCap的查询接口来获取到用户设备当前的设备SysCap集合(即步骤S902)。

[0192] 参阅图10,在本申请实施例中,一方面,Samgr部件可以为系统各个部件(例如部件A、部件B等等)提供注册接口和查询接口,例如,系统可以向Samgr部件发送注册部件(RegSysCap)的请求,Samgr部件可以执行注册过程,之后再返回注册结果(true/false);再例如,系统可以向Samgr部件发送查询部件(QueSysCap)的请求,Samgr部件可以查询相应部件,之后再返回查询结果(true/false)。另一方面,Samgr部件还可以为应用(Application, APP)提供SysCap的查询接口,应用可以在调用某个API之前检测系统是否具备与该API相关联的SysCap,例如,应用需要调用某个API,则应用可以先向Samgr部件发送查询与该API相关联的部件(hasSysCap)的请求,Samgr部件可以查询相应部件,之后再返回查询结果(true/false),若查询结果为true,则表示系统具备与该API相关联的SysCap,进一步地,应用可以调用该API执行相应功能。

[0193] 用户设备的PMS在通过调用系统管理服务模块中Samgr部件提供的SysCap的查询接口获取到用户设备当前的设备SysCap集合之后,可以判断应用SysCap集合的范围是否小于或等于用户设备当前的设备SysCap集合的范围(即步骤S903),若是,则用户设备的PMS可以确认应用可以在该用户设备上安装并正常运行(即步骤S904),若否,则用户设备的PMS可以确认应用不可以在该用户设备上安装(即步骤S905)。

[0194] 示例性地,在步骤S722中,应用SysCap集合为应用SysCap集合1,则用户设备的PMS可以判断应用SysCap集合1的范围是否小于或等于用户设备当前的设备SysCap集合的范围,若是,则确认应用SysCap集合1对应的应用可以在该用户设备上安装并正常运行,也即是说,在应用SysCap集合1的范围小于或等于用户设备当前的设备SysCap集合的范围的情

况下,该用户设备上可以成功安装并正常运行应用SysCap集合1对应的应用。

[0195] 示例性地,如图11A所示,图11A所示的用户界面可以为用户设备上应用市场(也可以称为应用商店)这一应用的“推荐”界面,该界面中可以显示有一个或多个向用户推荐的应用的信息(例如应用1的图标、应用2的图标、应用3图标、应用4图标等信息)。其中,这些应用的应用SysCap集合的范围均小于或等于该用户设备的设备SysCap集合。

[0196] 也即是说,只有当某个应用的应用SysCap集合的范围小于或等于用户设备的设备SysCap集合,该应用才可能会被显示在应用市场的“推荐”界面中并支持用户来安装该应用到该用户设备。

[0197] 进一步地,如果用户想要安装某个应用,例如,应用1,则用户可以点击应用1对应的安装选项,在安装成功后,用户设备可以在图11B中显示应用1对应的打开选项,用户点击打开选项即可打开该应用。

[0198] 示例性地,如图11C所示,图11C所示的用户界面可以为用户设备上应用市场这一应用的“搜索”界面,如果用户想要在用户设备上安装某个应用,例如,XX应用,则用户可以在图11C所示的搜索框中输入“XX应用”。

[0199] 如果XX应用的应用SysCap集合的范围小于或等于用户设备的设备SysCap集合,则用户设备可以在图11C所示的用户界面中显示该应用可以在该用户设备上安装的信息,例如,显示该应用对应的安装选项,用户点击安装选项即可安装该应用。

[0200] 如果XX应用的应用SysCap集合的范围大于用户设备的设备SysCap集合,则用户设备可以在图11D所示的用户界面中显示该应用无法在该用户设备上安装的信息,例如,可以灰度化显示该应用的图标及该应用的名称、以及该应用对应的无法安装的选项,用于提示用户该应用无法安装到该用户设备上。

[0201] 在一些实施例中,由于应用分发平台在分发应用前已经确认应用SysCap集合1的范围小于或等于设备SysCap集合1的范围,因此,步骤S722是可选的,也即是说,用户设备无需再判断应用SysCap集合1的范围是否小于或等于当前设备SysCap集合的范围。

[0202] 在一些实施例中,在应用SysCap集合1的范围大于用户设备当前的设备SysCap集合的范围的情况下,应用SysCap集合1对应的应用可以通过免安装流程使得此应用在该用户设备上正常运行。

[0203] 通过实施图7所示实施例提供的方法,可以利用PCID和RPCID来解决OS弹性部署后应用与设备之间的兼容性问题,此外,在设备发布之前,基于设备认证平台中存储的PCID就可以获取到应用分发平台上提供的该设备可用的应用的数量,而且,可以提前让应用开发者进行核心应用(例如系统应用)的开发,缩短设备上市的时间。

[0204] 需要说明的是,图7所示实施例中各个阶段可以独立存在,即该阶段中的相应步骤可以是独立于其他阶段的步骤执行的;也可以某几个阶段自由组合,即这几个阶段中的相应步骤可以是独立于其他阶段的步骤执行的,本申请实施例对此不作限定。

[0205] 在本申请实施例中,第一标识可以是PCID,第二标识可以是RPCID,第一设备SysCap集合可以是设备SysCap集合1,第一应用SysCap集合可以是应用SysCap集合1,第一请求可以是应用安装请求,第一应用可以是应用SysCap集合1对应的应用,第一PCID可以是PCID1,第一RPCID可以是RPCID1,第一SDK可以是SDK1,第一信息可以是SDK1或PCID1,第一类型的用户设备可以包括但不限于手机、PC、平板、智慧屏、音箱、手表、眼镜、车机、耳机,第

二请求可以是设备认证请求1,第一令牌Token可以是Token1,第三请求可以是获取PCID的请求,第一用户界面可以是图11A或图11C所示的用户界面,第一选项可以是图11A或图11C所示的安装选项。

[0206] 下面介绍本申请实施例提供的一种用户设备(也可以称为电子设备100)的软件结构。

[0207] 图12性示出了本申请实施例中提供的一种电子设备100的软件结构。

[0208] 如图12所示,电子设备100的软件系统可以采用分层架构,事件驱动架构,微核架构,微服务架构,或云架构。本申请实施例以分层架构的Android系统为例,示例性说明电子设备100的软件结构。

[0209] 分层架构将软件分成若干个层,每一层都有清晰的角色和分工。层与层之间通过软件接口通信。在一些实施例中,将Android系统分为四层,从上至下分别为应用程序层,应用程序框架层,安卓运行时(Android runtime)和系统库,以及内核层。

[0210] 应用程序层可以包括一系列应用程序包。

[0211] 如图12所示,应用程序包可以包括相机,图库,日历,通话,地图,应用市场,WLAN,蓝牙,音乐,视频,短信息等应用程序。

[0212] 其中,应用市场可以用于下载安装应用程序。

[0213] 需要说明的是,应用市场应用程序的名称仅为本申请实施例所使用的词语,其代表的含义在本申请实施例中已经记载,其名称并不能对本申请实施例构成任何限制。

[0214] 应用程序框架层为应用程序层的应用程序提供应用编程接口(application programming interface,API)和编程框架。应用程序框架层包括一些预先定义的函数。

[0215] 如图12所示,应用程序框架层可以包括窗口管理器,内容提供器,视图系统,电话管理器,资源管理器,通知管理等。

[0216] 窗口管理器用于管理窗口程序。窗口管理器可以获取显示屏大小,判断是否有状态栏,锁定屏幕,截取屏幕等。

[0217] 内容提供器用来存放和获取数据,并使这些数据可以被应用程序访问。所述数据可以包括视频,图像,音频,拨打和接听的电话,浏览历史和书签,电话簿等。

[0218] 视图系统包括可视控件,例如显示文字的控件,显示图片的控件等。视图系统可用于构建应用程序。显示界面可以由一个或多个视图组成的。例如,包括短信通知图标的显示界面,可以包括显示文字的视图以及显示图片的视图。

[0219] 电话管理器用于提供电子设备100的通信功能。例如通话状态的管理(包括接通,挂断等)。

[0220] 资源管理器为应用程序提供各种资源,比如本地化字符串,图标,图片,布局文件,视频文件等等。

[0221] 通知管理器使应用程序可以在状态栏中显示通知信息,可以用于传达告知类型的消息,可以短暂停留后自动消失,无需用户交互。比如通知管理器被用于告知下载完成,消息提醒等。通知管理器还可以是以图表或者滚动条文本形式出现在系统顶部状态栏的通知,例如后台运行的应用程序的通知,还可以是以对话框形式出现在屏幕上的通知。例如在状态栏提示文本信息,发出提示音,电子设备振动,指示灯闪烁等。

[0222] Android Runtime包括核心库和虚拟机。Android runtime负责安卓系统的调度和

管理。

[0223] 核心库包含两部分：一部分是java语言需要调用的功能函数，另一部分是安卓的核心库。

[0224] 应用程序层和应用程序框架层运行在虚拟机中。虚拟机将应用程序层和应用程序框架层的java文件执行为二进制文件。虚拟机用于执行对象生命周期的管理，堆栈管理，线程管理，安全和异常的管理，以及垃圾回收等功能。

[0225] 系统库可以包括多个功能模块。例如：表面管理器(surface manager)，媒体库(Media Libraries)，三维图形处理库(例如：OpenGL ES)，2D图形引擎(例如：SGL)等。

[0226] 表面管理器用于对显示子系统进行管理，并且为多个应用程序提供了2D和3D图层的融合。

[0227] 媒体库支持多种常用的音频，视频格式回放和录制，以及静态图像文件等。媒体库可以支持多种音视频编码格式，例如：MPEG4，H.264，MP3，AAC，AMR，JPG，PNG等。

[0228] 三维图形处理库用于实现三维图形绘图，图像渲染，合成，和图层处理等。

[0229] 2D图形引擎是2D绘图的绘图引擎。

[0230] 内核层是硬件和软件之间的层。内核层至少包含显示驱动，摄像头驱动，蓝牙驱动，传感器驱动。

[0231] 在本申请实施例中，电子设备100的软件架构中还可以包括包管理服务模块(图中未示出)和系统服务管理模块(图中未示出)，其中，这两个模块的具体功能以及工作细节可以参照前述各个实施例中的相关内容，在此不再赘述。

[0232] 下面结合捕获拍照场景，示例性说明电子设备100软件以及硬件的工作流程。

[0233] 当触摸传感器180K接收到触摸操作，相应的硬件中断被发给内核层。内核层将触摸操作加工成原始输入事件(包括触摸坐标，触摸操作的时间戳等信息)。原始输入事件被存储在内核层。应用程序框架层从内核层获取原始输入事件，识别该输入事件所对应的控件。以该触摸操作是触摸单击操作，该单击操作所对应的控件为相机应用图标的控件为例，相机应用调用应用框架层的接口，启动相机应用，进而通过调用内核层启动摄像头驱动，通过摄像头193捕获静态图像或视频。

[0234] 下面介绍本申请实施例提供的一种用户设备(也可以称为电子设备100)的结构。

[0235] 图13示例性示出了本申请实施例中提供的一种电子设备100的结构。

[0236] 如图13所示，电子设备100可以包括：处理器110，外部存储器接口120，内部存储器121，通用串行总线(universal serial bus,USB)接口130，充电管理模块140，电源管理模块141，电池142，天线1，天线2，移动通信模块150，无线通信模块160，音频模块170，扬声器170A，受话器170B，麦克风170C，耳机接口170D，传感器模块180，按键190，马达191，指示器192，摄像头193，显示屏194，以及用户标识模块(subscriber identification module, SIM)卡接口195等。其中传感器模块180可以包括压力传感器180A，陀螺仪传感器180B，气压传感器180C，磁传感器180D，加速度传感器180E，距离传感器180F，接近光传感器180G，指纹传感器180H，温度传感器180J，触摸传感器180K，环境光传感器180L，骨传导传感器180M等。

[0237] 可以理解的是，本申请实施例示意的结构并不构成对电子设备100的具体限定。在本申请另一些实施例中，电子设备100可以包括比图示更多或更少的部件，或者组合某些部件，或者拆分某些部件，或者不同的部件布置。图示的部件可以以硬件，软件或软件和硬件

的组合实现。

[0238] 处理器110可以包括一个或多个处理单元,例如:处理器110可以包括应用处理器(application processor,AP),调制解调处理器,图形处理器(graphics processing unit,GPU),图像信号处理器(image signal processor,ISP),控制器,存储器,视频编解码器,数字信号处理器(digital signal processor,DSP),基带处理器,和/或神经网络处理器(neural-network processing unit,NPU)等。其中,不同的处理单元可以是独立的器件,也可以集成在一个或多个处理器中。

[0239] 其中,控制器可以是电子设备100的神经中枢和指挥中心。控制器可以根据指令操作码和时序信号,产生操作控制信号,完成取指令和执行指令的控制。

[0240] 处理器110中还可以设置存储器,用于存储指令和数据。在一些实施例中,处理器110中的存储器为高速缓冲存储器。该存储器可以保存处理器110刚用过或循环使用的指令或数据。如果处理器110需要再次使用该指令或数据,可从所述存储器中直接调用。避免了重复存取,减少了处理器110的等待时间,因而提高了系统的效率。

[0241] 在一些实施例中,处理器110可以包括一个或多个接口。接口可以包括集成电路(inter-integrated circuit,I2C)接口,集成电路内置音频(inter-integrated circuit sound,I2S)接口,脉冲编码调制(pulse code modulation,PCM)接口,通用异步收发传输器(universal asynchronous receiver/transmitter,UART)接口,移动产业处理器接口(mobile industry processor interface,MIPI),通用输入输出(general-purpose input/output,GPIO)接口,用户标识模块(subscriber identity module,SIM)接口,和/或通用串行总线(universal serial bus,USB)接口等。

[0242] I2C接口是一种双向同步串行总线,包括一根串行数据线(serial data line,SDA)和一根串行时钟线(derail clock line,SCL)。在一些实施例中,处理器110可以包含多组I2C总线。处理器110可以通过不同的I2C总线接口分别耦合触摸传感器180K,充电器,闪光灯,摄像头193等。例如:处理器110可以通过I2C接口耦合触摸传感器180K,使处理器110与触摸传感器180K通过I2C总线接口通信,实现电子设备100的触摸功能。

[0243] I2S接口可以用于音频通信。在一些实施例中,处理器110可以包含多组I2S总线。处理器110可以通过I2S总线与音频模块170耦合,实现处理器110与音频模块170之间的通信。在一些实施例中,音频模块170可以通过I2S接口向无线通信模块160传递音频信号,实现通过蓝牙耳机接听电话的功能。

[0244] PCM接口也可以用于音频通信,将模拟信号抽样,量化和编码。在一些实施例中,音频模块170与无线通信模块160可以通过PCM总线接口耦合。在一些实施例中,音频模块170也可以通过PCM接口向无线通信模块160传递音频信号,实现通过蓝牙耳机接听电话的功能。所述I2S接口和所述PCM接口都可以用于音频通信。

[0245] UART接口是一种通用串行数据总线,用于异步通信。该总线可以为双向通信总线。它将要传输的数据在串行通信与并行通信之间转换。在一些实施例中,UART接口通常被用于连接处理器110与无线通信模块160。例如:处理器110通过UART接口与无线通信模块160中的蓝牙模块通信,实现蓝牙功能。在一些实施例中,音频模块170可以通过UART接口向无线通信模块160传递音频信号,实现通过蓝牙耳机播放音乐的功能。

[0246] MIPI接口可以被用于连接处理器110与显示屏194,摄像头193等外围器件。MIPI接

口包括摄像头串行接口(camera serial interface,CSI),显示屏串行接口(display serial interface,DSI)等。在一些实施例中,处理器110和摄像头193通过CSI接口通信,实现电子设备100的拍摄功能。处理器110和显示屏194通过DSI接口通信,实现电子设备100的显示功能。

[0247] GPIO接口可以通过软件配置。GPIO接口可以被配置为控制信号,也可被配置为数据信号。在一些实施例中,GPIO接口可以用于连接处理器110与摄像头193,显示屏194,无线通信模块160,音频模块170,传感器模块180等。GPIO接口还可以被配置为I2C接口,I2S接口,UART接口,MIPI接口等。

[0248] USB接口130是符合USB标准规范的接口,具体可以是Mini USB接口,Micro USB接口,USB Type C接口等。USB接口130可以用于连接充电器为电子设备100充电,也可以用于电子设备100与外围设备之间传输数据。也可以用于连接耳机,通过耳机播放音频。该接口还可以用于连接其他终端设备,例如AR设备等。

[0249] 可以理解的是,本申请实施例示意的各模块间的接口连接关系,只是示意性说明,并不构成对电子设备100的结构限定。在本申请另一些实施例中,电子设备100也可以采用上述实施例中不同的接口连接方式,或多种接口连接方式的组合。

[0250] 充电管理模块140用于从充电器接收充电输入。其中,充电器可以是无线充电器,也可以是有线充电器。在一些有线充电的实施例中,充电管理模块140可以通过USB接口130接收有线充电器的充电输入。在一些无线充电的实施例中,充电管理模块140可以通过电子设备100的无线充电线圈接收无线充电输入。充电管理模块140为电池142充电的同时,还可以通过电源管理模块141为电子设备100供电。

[0251] 电源管理模块141用于连接电池142,充电管理模块140与处理器110。电源管理模块141接收电池142和/或充电管理模块140的输入,为处理器110,内部存储器121,外部存储器,显示屏194,摄像头193,和无线通信模块160等供电。电源管理模块141还可以用于监测电池容量,电池循环次数,电池健康状态(漏电,阻抗)等参数。在其他一些实施例中,电源管理模块141也可以设置于处理器110中。在另一些实施例中,电源管理模块141和充电管理模块140也可以设置于同一个器件中。

[0252] 电子设备100的无线通信功能可以通过天线1,天线2,移动通信模块150,无线通信模块160,调制解调处理器以及基带处理器等实现。

[0253] 天线1和天线2用于发射和接收电磁波信号。电子设备100中的每个天线可用于覆盖单个或多个通信频带。不同的天线还可以复用,以提高天线的利用率。例如:可以将天线1复用为无线局域网的分集天线。在另外一些实施例中,天线可以和调谐开关结合使用。

[0254] 移动通信模块150可以提供应用在电子设备100上的包括2G/3G/4G/5G等无线通信的解决方案。移动通信模块150可以包括至少一个滤波器,开关,功率放大器,低噪声放大器(low noise amplifier,LNA)等。移动通信模块150可以由天线1接收电磁波,并对接收的电磁波进行滤波,放大等处理,传送至调制解调处理器进行解调。移动通信模块150还可以对经调制解调处理器调制后的信号放大,经天线1转为电磁波辐射出去。在一些实施例中,移动通信模块150的至少部分功能模块可以被设置于处理器110中。在一些实施例中,移动通信模块150的至少部分功能模块可以与处理器110的至少部分模块被设置在同一个器件中。

[0255] 调制解调处理器可以包括调制器和解调器。其中,调制器用于将待发送的低频基

带信号调制成中高频信号。解调器用于将接收的电磁波信号解调为低频基带信号。随后解调器将解调得到的低频基带信号传送至基带处理器处理。低频基带信号经基带处理器处理后,被传递给应用处理器。应用处理器通过音频设备(不限于扬声器170A,受话器170B等)输出声音信号,或通过显示屏194显示图像或视频。在一些实施例中,调制解调处理器可以是独立的器件。在另一些实施例中,调制解调处理器可以独立于处理器110,与移动通信模块150或其他功能模块设置在同一个器件中。

[0256] 无线通信模块160可以提供应用在电子设备100上的包括无线局域网(wireless local area networks,WLAN)(如无线保真(wireless fidelity,Wi-Fi)网络),蓝牙(bluetooth,BT),全球导航卫星系统(global navigation satellite system,GNSS),调频(frequency modulation,FM),近距离无线通信技术(near field communication,NFC),红外技术(infrared,IR)等无线通信的解决方案。无线通信模块160可以是集成至少一个通信处理模块的一个或多个器件。无线通信模块160经由天线2接收电磁波,将电磁波信号调频以及滤波处理,将处理后的信号发送到处理器110。无线通信模块160还可以从处理器110接收待发送的信号,对其进行调频,放大,经天线2转为电磁波辐射出去。

[0257] 在一些实施例中,电子设备100的天线1和移动通信模块150耦合,天线2和无线通信模块160耦合,使得电子设备100可以通过无线通信技术与网络以及其他设备通信。所述无线通信技术可以包括全球移动通讯系统(global system for mobile communications,GSM),通用分组无线服务(general packet radio service,GPRS),码分多址接入(code division multiple access,CDMA),宽带码分多址(wideband code division multiple access,WCDMA),时分码分多址(time-division code division multiple access,TD-SCDMA),长期演进(long term evolution,LTE),BT,GNSS,WLAN,NFC,FM,和/或IR技术等。所述GNSS可以包括全球卫星定位系统(global positioning system,GPS),全球导航卫星系统(global navigation satellite system,GLONASS),北斗卫星导航系统(beidou navigation satellite system,BDS),准天顶卫星系统(quasi-zenith satellite system,QZSS)和/或星基增强系统(satellite based augmentation systems,SBAS)。

[0258] 电子设备100通过GPU,显示屏194,以及应用处理器等实现显示功能。GPU为图像处理的微处理器,连接显示屏194和应用处理器。GPU用于执行数学和几何计算,用于图形渲染。处理器110可包括一个或多个GPU,其执行程序指令以生成或改变显示信息。

[0259] 显示屏194用于显示图像,视频等。显示屏194包括显示面板。显示面板可以采用液晶显示屏(liquid crystal display,LCD),有机发光二极管(organic light-emitting diode,OLED),有源矩阵有机发光二极管或主动矩阵有机发光二极管(active-matrix organic light emitting diode的,AMOLED),柔性发光二极管(flex light-emitting diode,FLED),Miniled,MicroLed,Micro-oLed,量子点发光二极管(quantum dot light emitting diodes,QLED)等。在一些实施例中,电子设备100可以包括1个或N个显示屏194,N为大于1的正整数。

[0260] 电子设备100可以通过ISP,摄像头193,视频编解码器,GPU,显示屏194以及应用处理器等实现拍摄功能。

[0261] ISP用于处理摄像头193反馈的数据。例如,拍照时,打开快门,光线通过镜头被传递到摄像头感光元件上,光信号转换为电信号,摄像头感光元件将所述电信号传递给ISP处

理,转化为肉眼可见的图像。ISP还可以对图像的噪点,亮度,肤色进行算法优化。ISP还可以对拍摄场景的曝光,色温等参数优化。在一些实施例中,ISP可以设置在摄像头193中。

[0262] 摄像头193用于捕获静态图像或视频。物体通过镜头生成光学图像投射到感光元件。感光元件可以是电荷耦合器件(charge coupled device,CCD)或互补金属氧化物半导体(complementary metal-oxide-semiconductor,CMOS)光电晶体管。感光元件把光信号转换成电信号,之后将电信号传递给ISP转换成数字图像信号。ISP将数字图像信号输出到DSP加工处理。DSP将数字图像信号转换成标准的RGB,YUV等格式的图像信号。在一些实施例中,电子设备100可以包括1个或N个摄像头193,N为大于1的正整数。

[0263] 数字信号处理器用于处理数字信号,除了可以处理数字图像信号,还可以处理其他数字信号。例如,当电子设备100在频点选择时,数字信号处理器用于对频点能量进行傅里叶变换等。

[0264] 视频编解码器用于对数字视频压缩或解压缩。电子设备100可以支持一种或多种视频编解码器。这样,电子设备100可以播放或录制多种编码格式的视频,例如:动态图像专家组(moving picture experts group,MPEG)1,MPEG2,MPEG3,MPEG4等。

[0265] NPU为神经网络(neural-network,NN)计算处理器,通过借鉴生物神经网络结构,例如借鉴人脑神经元之间传递模式,对输入信息快速处理,还可以不断的自学习。通过NPU可以实现电子设备100的智能认知等应用,例如:图像识别,人脸识别,语音识别,文本理解等。

[0266] 外部存储器接口120可以用于连接外部存储卡,例如Micro SD卡,实现扩展电子设备100的存储能力。外部存储卡通过外部存储器接口120与处理器110通信,实现数据存储功能。例如将音乐,视频等文件保存在外部存储卡中。

[0267] 内部存储器121可以用于存储计算机可执行程序代码,所述可执行程序代码包括指令。处理器110通过运行存储在内部存储器121的指令,从而执行电子设备100的各种功能应用以及数据处理。内部存储器121可以包括存储程序区和存储数据区。其中,存储程序区可存储操作系统,至少一个功能所需的应用程序(比如声音播放功能,图像播放功能等)等。存储数据区可存储电子设备100使用过程中所创建的数据(比如音频数据,电话本等)等。此外,内部存储器121可以包括高速随机存取存储器,还可以包括非易失性存储器,例如至少一个磁盘存储器件,闪存器件,通用闪存存储器(universal flash storage,UFS)等。

[0268] 电子设备100可以通过音频模块170,扬声器170A,受话器170B,麦克风170C,耳机接口170D,以及应用处理器等实现音频功能。例如音乐播放,录音等。

[0269] 音频模块170用于将数字音频信息转换成模拟音频信号输出,也用于将模拟音频输入转换为数字音频信号。音频模块170还可以用于对音频信号编码和解码。在一些实施例中,音频模块170可以设置于处理器110中,或将音频模块170的部分功能模块设置于处理器110中。

[0270] 扬声器170A,也称“喇叭”,用于将音频电信号转换为声音信号。电子设备100可以通过扬声器170A收听音乐,或收听免提通话。

[0271] 受话器170B,也称“听筒”,用于将音频电信号转换成声音信号。当电子设备100接听电话或语音信息时,可以通过将受话器170B靠近人耳接听语音。

[0272] 麦克风170C,也称“话筒”,“传声器”,用于将声音信号转换为电信号。当拨打电话

或发送语音信息时,用户可以通过人嘴靠近麦克风170C发声,将声音信号输入到麦克风170C。电子设备100可以设置至少一个麦克风170C。在另一些实施例中,电子设备100可以设置两个麦克风170C,除了采集声音信号,还可以实现降噪功能。在另一些实施例中,电子设备100还可以设置三个,四个或更多麦克风170C,实现采集声音信号,降噪,还可以识别声音来源,实现定向录音功能等。

[0273] 耳机接口170D用于连接有线耳机。耳机接口170D可以是USB接口130,也可以是3.5mm的开放移动终端设备平台(open mobile terminal platform,OMTP)标准接口,美国蜂窝电信工业协会(cellular telecommunications industry association of the USA,CTIA)标准接口。

[0274] 触摸传感器180K,也称“触控面板”。触摸传感器180K可以设置于显示屏194,由触摸传感器180K与显示屏194组成触摸屏,也称“触控屏”。触摸传感器180K用于检测作用于其上或附近的触摸操作。触摸传感器可以将检测到的触摸操作传递给应用处理器,以确定触摸事件类型。可以通过显示屏194提供与触摸操作相关的视觉输出。在另一些实施例中,触摸传感器180K也可以设置于电子设备100的表面,与显示屏194所处的位置不同。

[0275] 按键190包括开机键,音量键等。按键190可以是机械按键。也可以是触摸式按键。电子设备100可以接收按键输入,产生与电子设备100的用户设置以及功能控制有关的键信号输入。

[0276] 马达191可以产生振动提示。马达191可以用于来电振动提示,也可以用于触摸振动反馈。例如,作用于不同应用(例如拍照,音频播放等)的触摸操作,可以对应不同的振动反馈效果。作用于显示屏194不同区域的触摸操作,马达191也可对应不同的振动反馈效果。不同的应用场景(例如:时间提醒,接收信息,闹钟,游戏等)也可以对应不同的振动反馈效果。触摸振动反馈效果还可以支持自定义。

[0277] 指示器192可以是指示灯,可以用于指示充电状态,电量变化,也可以用于指示消息,未接来电,通知等。

[0278] SIM卡接口195用于连接SIM卡。SIM卡可以通过插入SIM卡接口195,或从SIM卡接口195拔出,实现和电子设备100的接触和分离。电子设备100可以支持1个或N个SIM卡接口,N为大于1的正整数。SIM卡接口195可以支持Nano SIM卡,Micro SIM卡,SIM卡等。同一个SIM卡接口195可以同时插入多张卡。所述多张卡的类型可以相同,也可以不同。SIM卡接口195也可以兼容不同类型的SIM卡。SIM卡接口195也可以兼容外部存储卡。电子设备100通过SIM卡和网络交互,实现通话以及数据通信等功能。在一些实施例中,电子设备100采用eSIM,即:嵌入式SIM卡。eSIM卡可以嵌在电子设备100中,不能和电子设备100分离。

[0279] 应当理解的是,图13所示电子设备100仅是一个范例,并且电子设备100可以具有比图13中所示的更多的或者更少的部件,可以组合两个或多个的部件,或者可以具有不同的部件配置。图13中所示出的各种部件可以在包括一个或多个信号处理和/或专用集成电路在内的硬件、软件、或硬件和软件的组合中实现。

[0280] 在上述实施例中,可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时,可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时,全部或部分地产生按照本申请所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计

算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中,或者从一个计算机可读存储介质向另一个计算机可读存储介质传输,例如,所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例如同轴电缆、光纤、数字用户线)或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质,(例如,软盘、硬盘、磁带)、光介质(例如,DVD)、或者半导体介质(例如固态硬盘(solid state disk,SSD))等。

[0281] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,该流程可以由计算机程序来指令相关的硬件完成,该程序可存储于计算机可读存储介质中,该程序在执行时,可包括如上述各方法实施例的流程。而前述的存储介质包括:ROM或随机存储记忆体RAM、磁碟或者光盘等各种可存储程序代码的介质。

[0282] 以上实施例仅用以说明本申请的技术方案,而非对其限制;尽管参照前述实施例对本申请进行了详细的说明,本领域的普通技术人员应当理解:其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换;而这些修改或者替换,并不使相应技术方案的本质脱离本申请各实施例技术方案的范围。

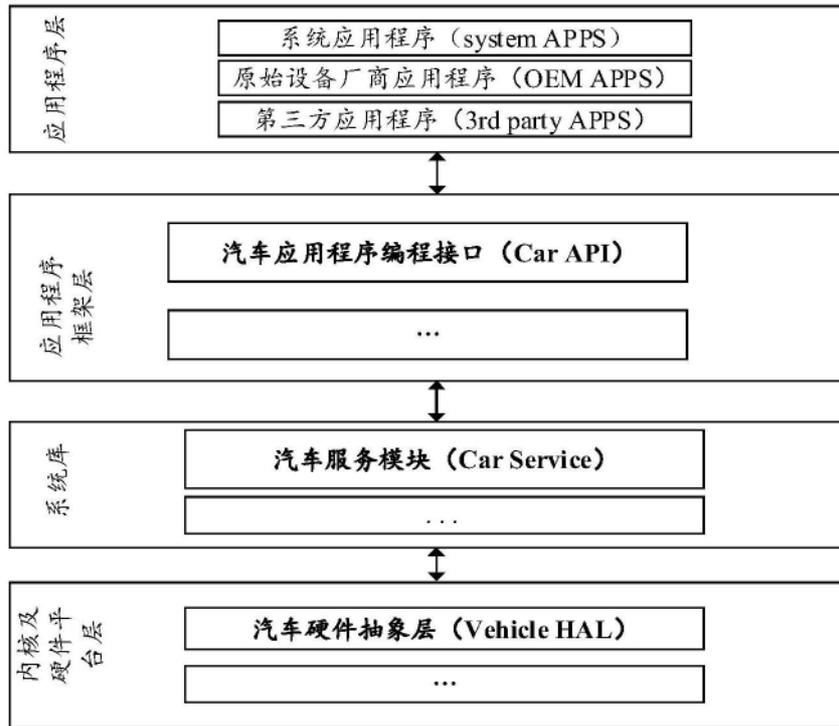


图1

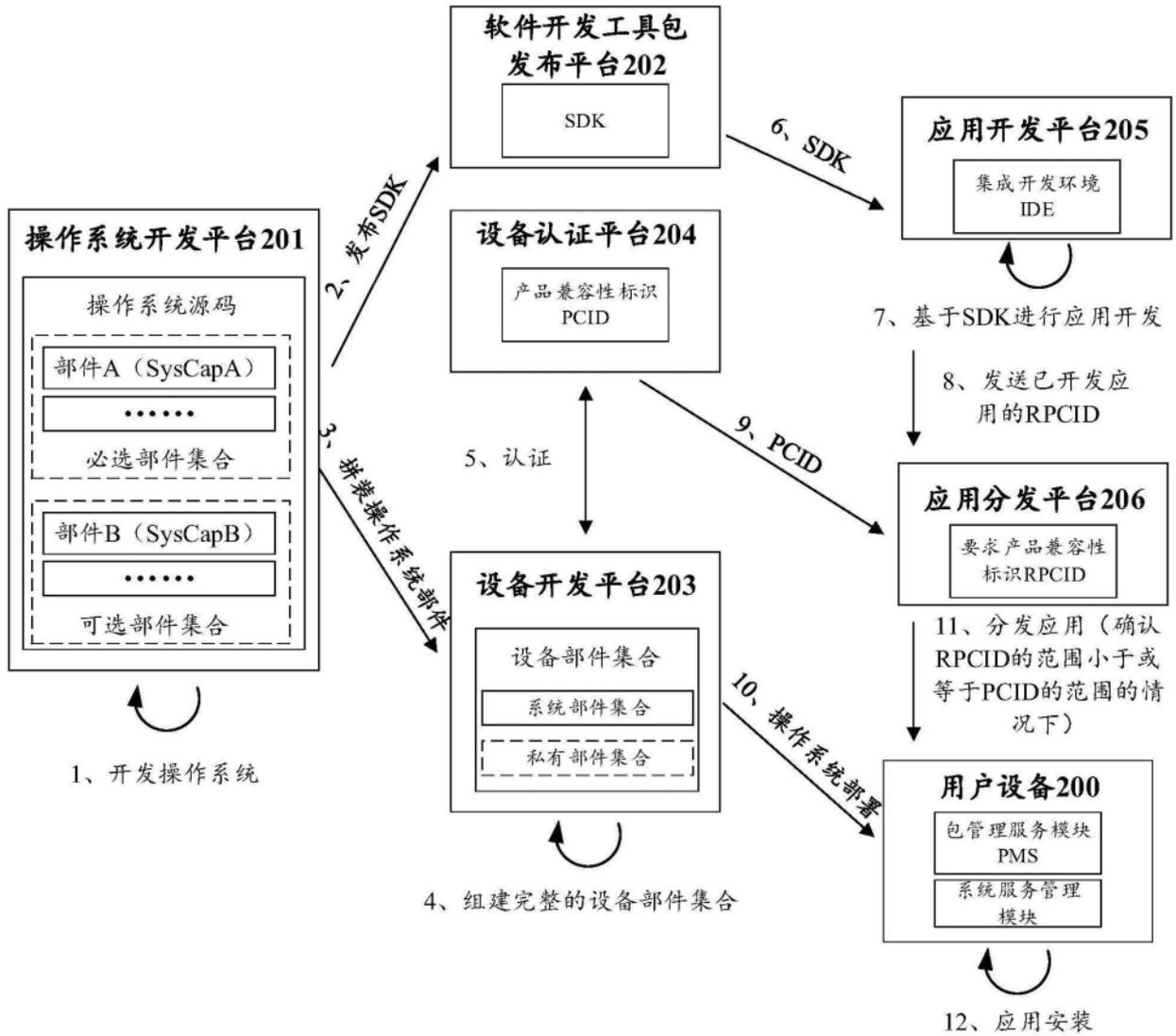


图2

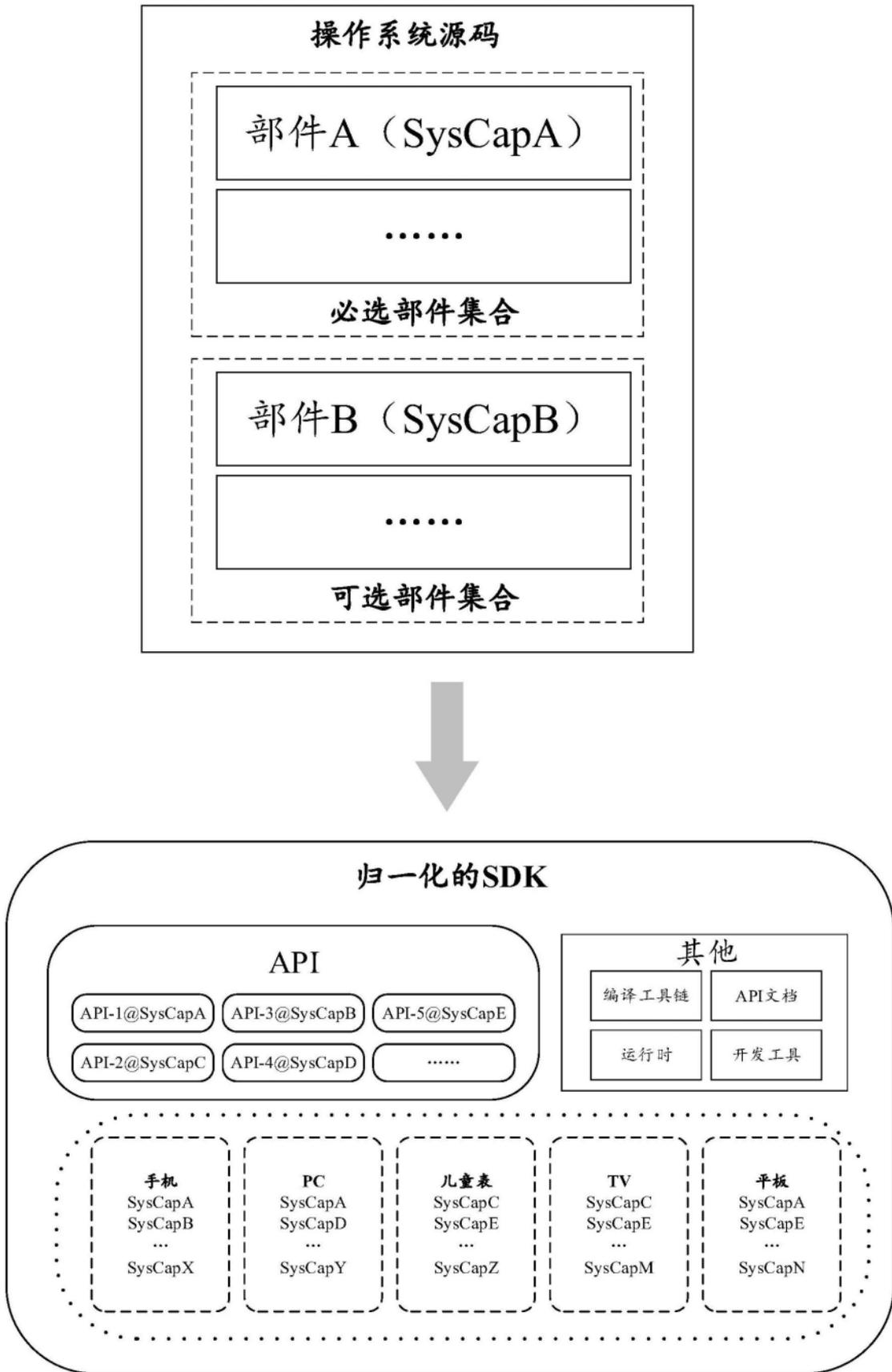


图3

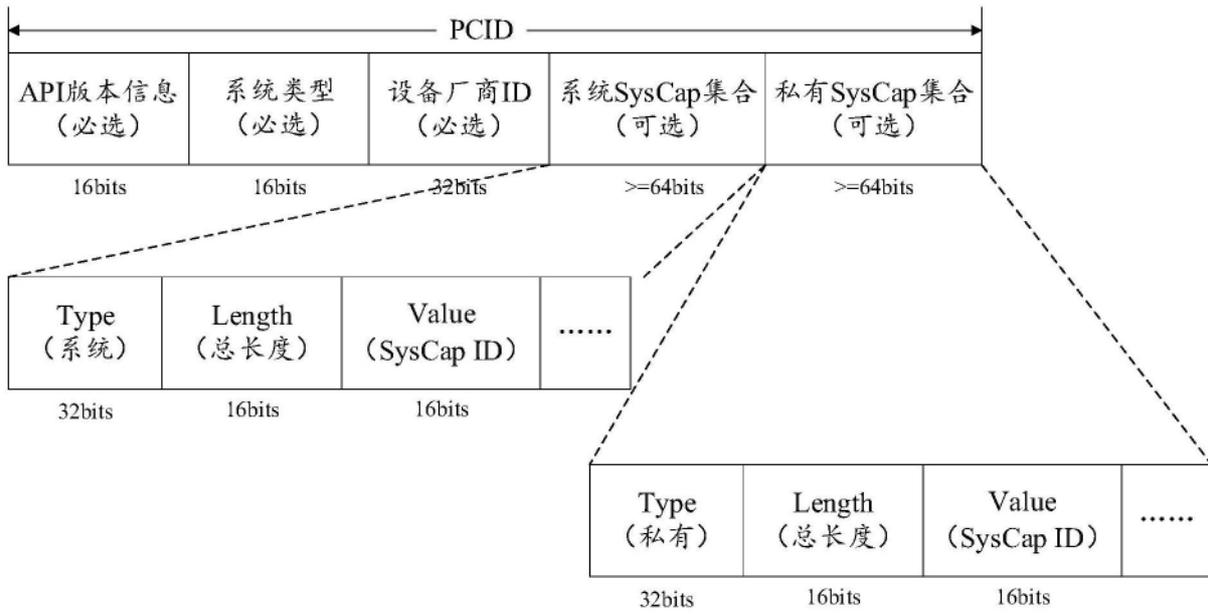


图4

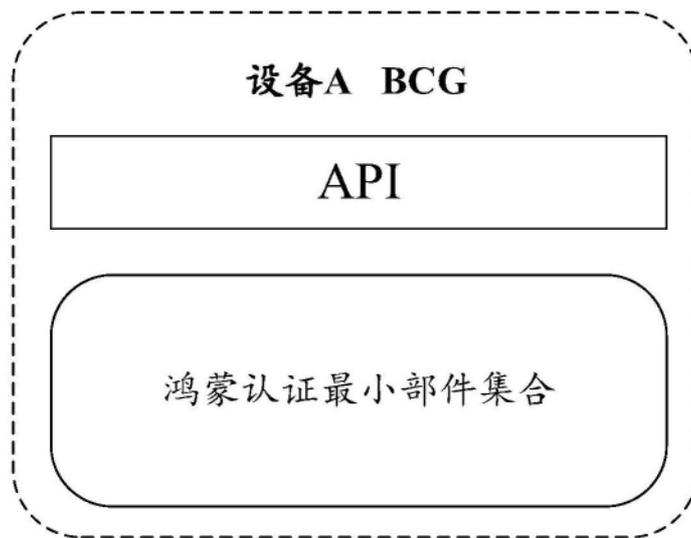


图5A

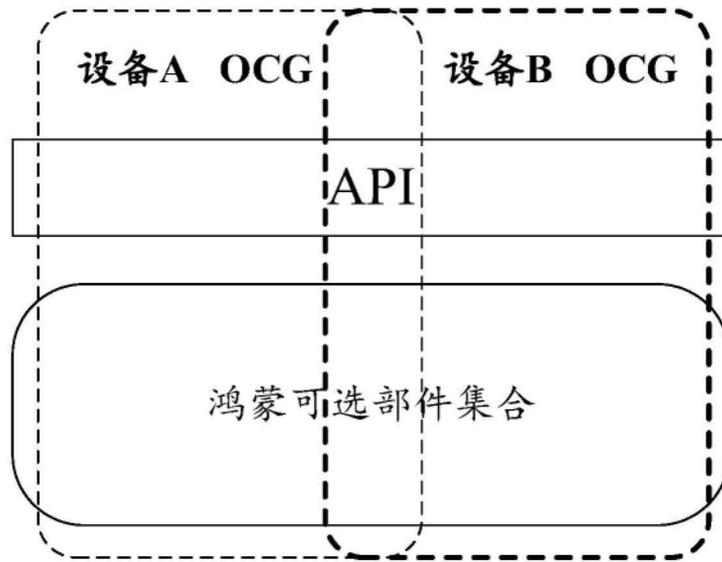


图5B

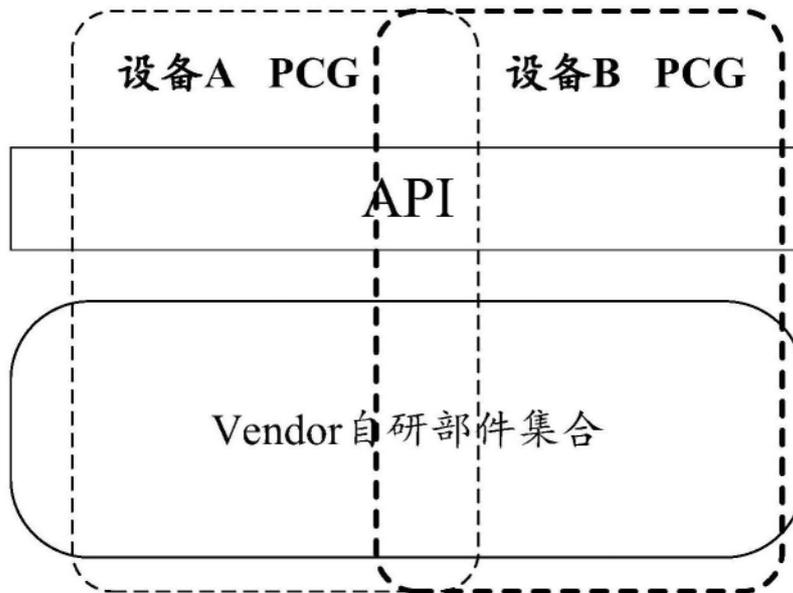


图5C

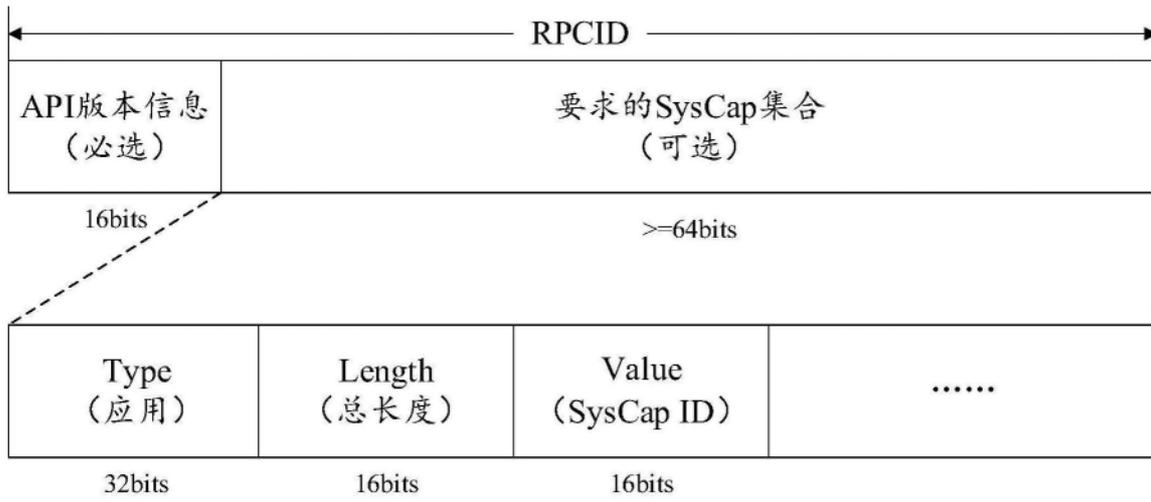


图6

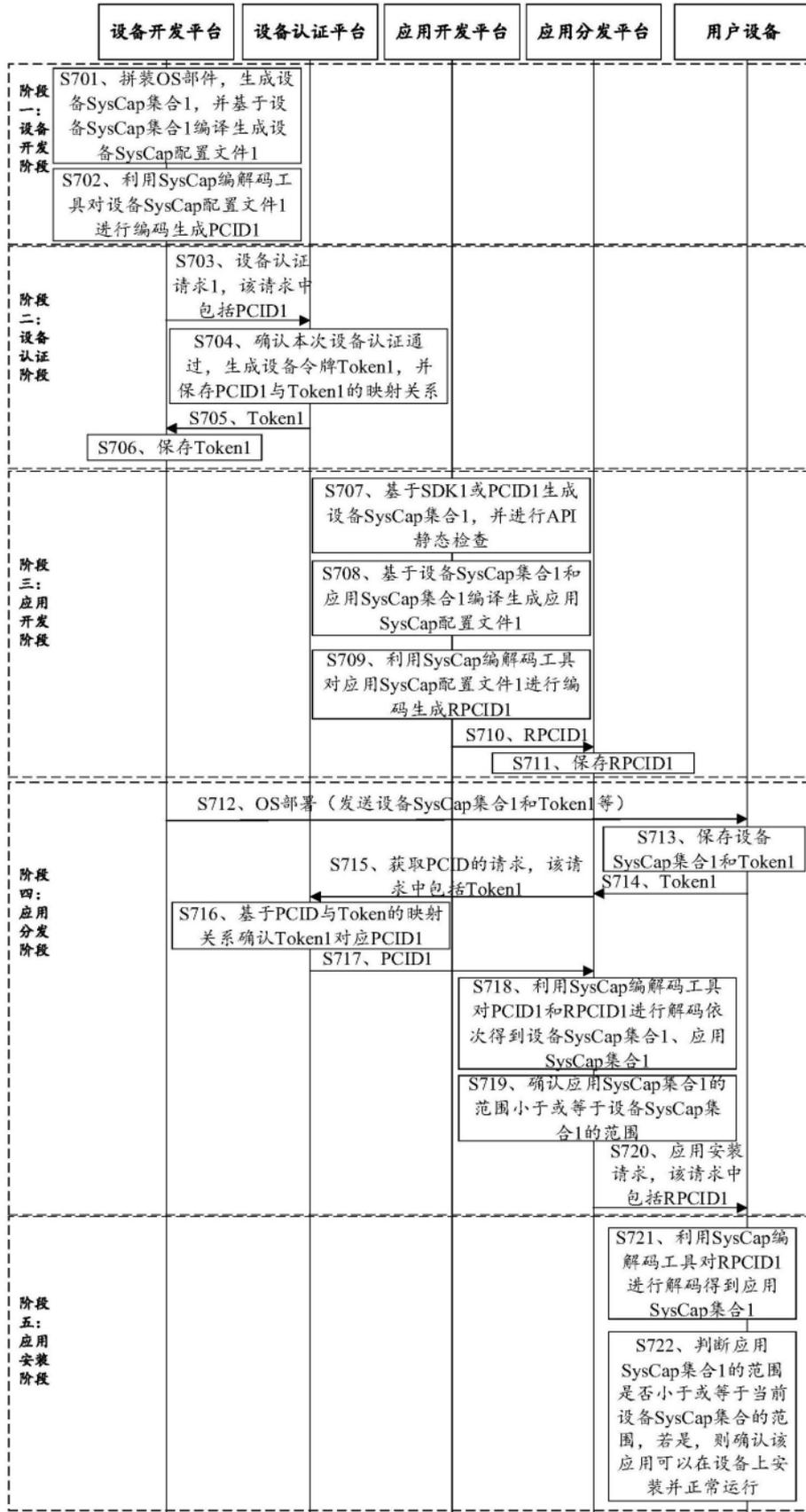


图7

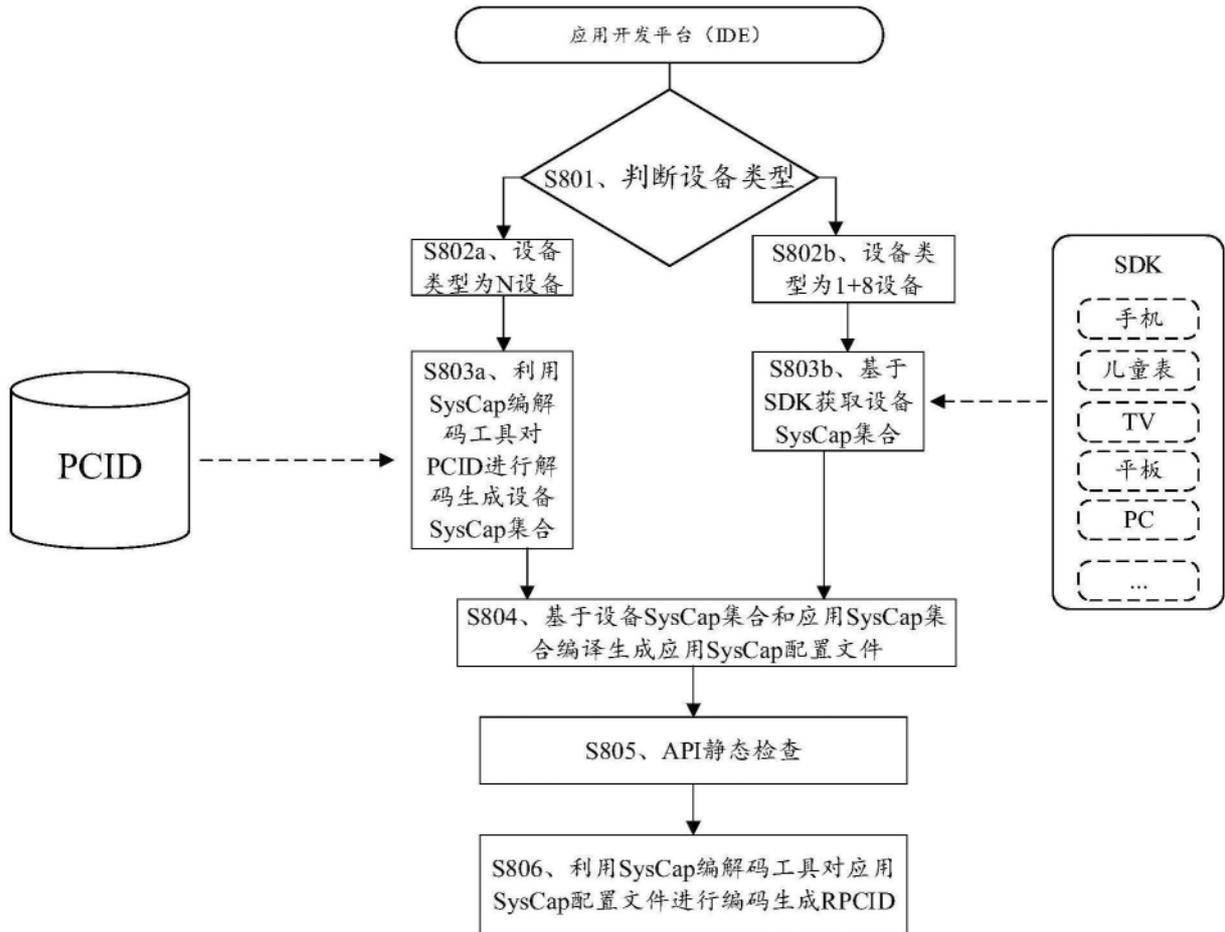


图8

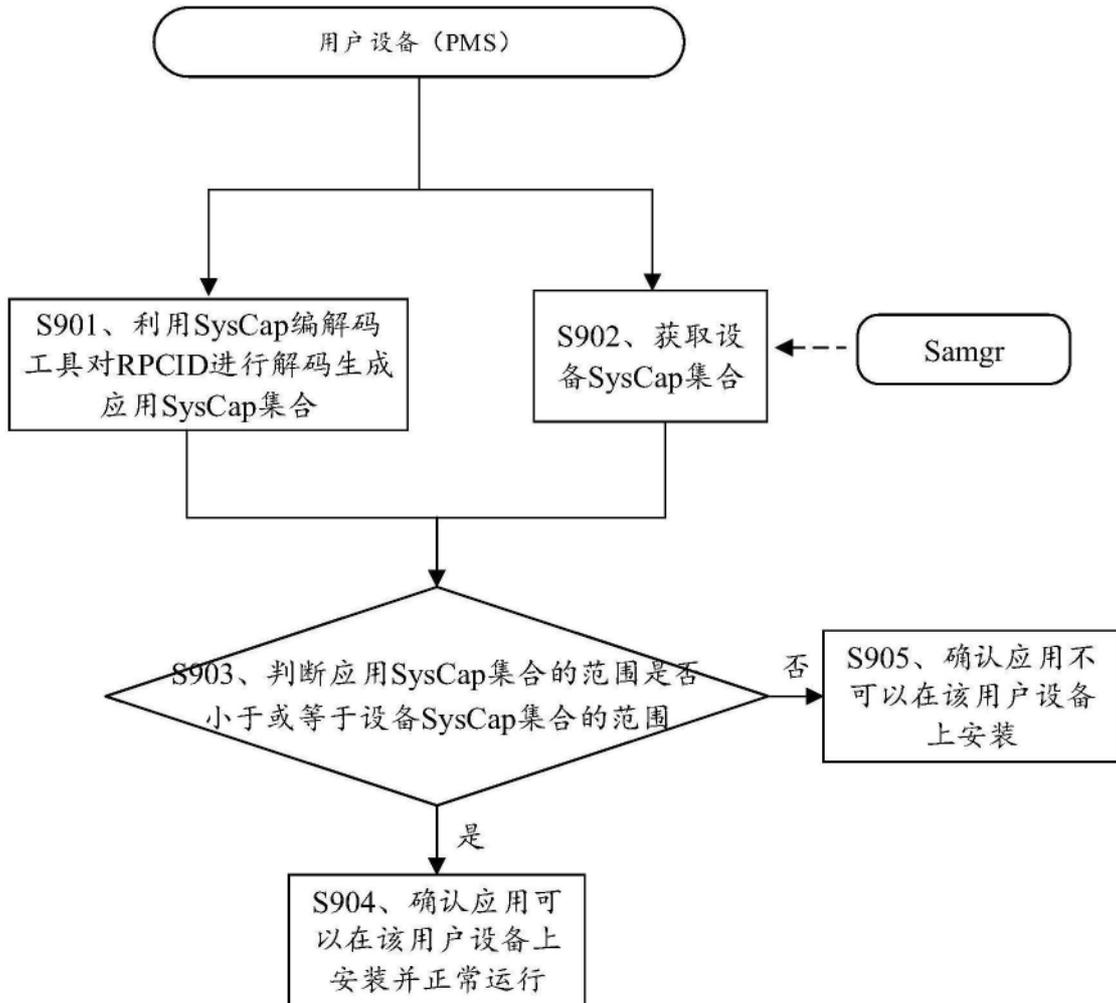


图9

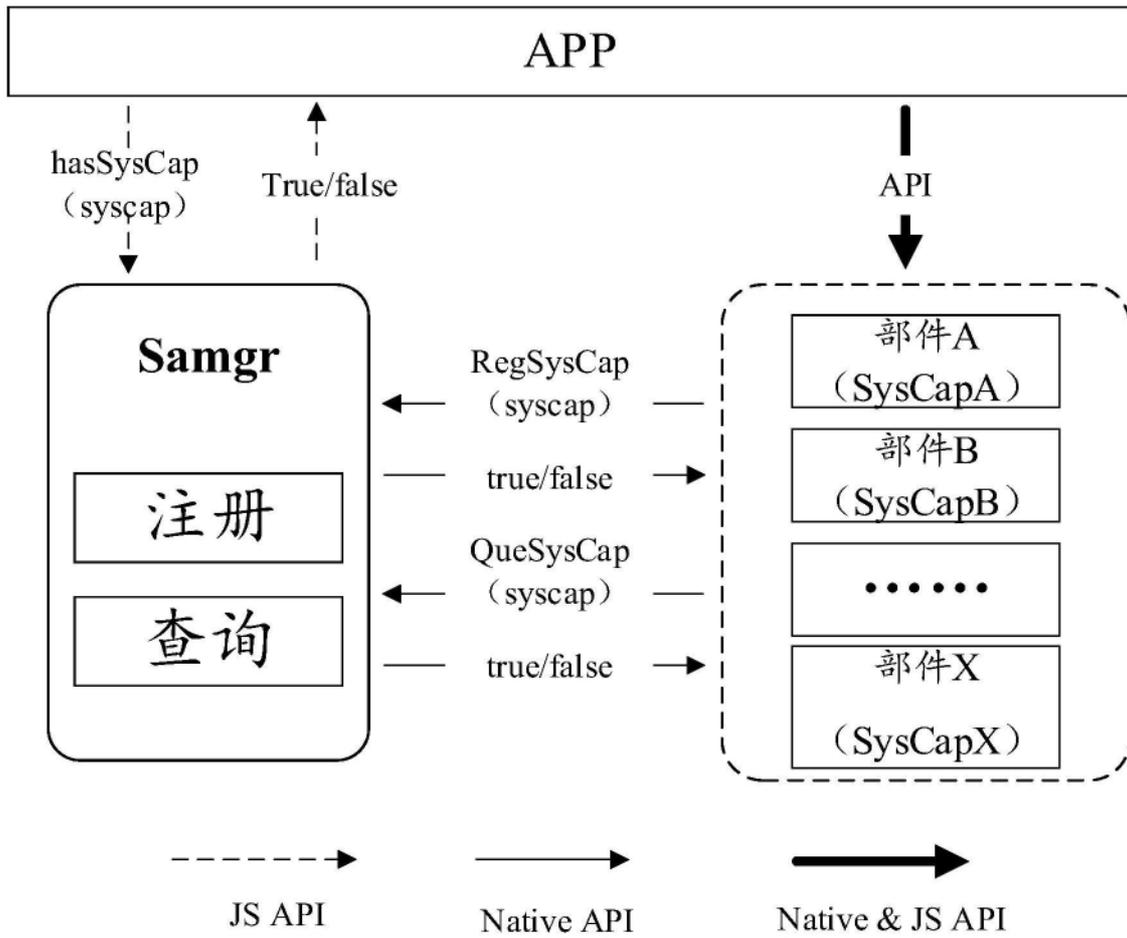


图10

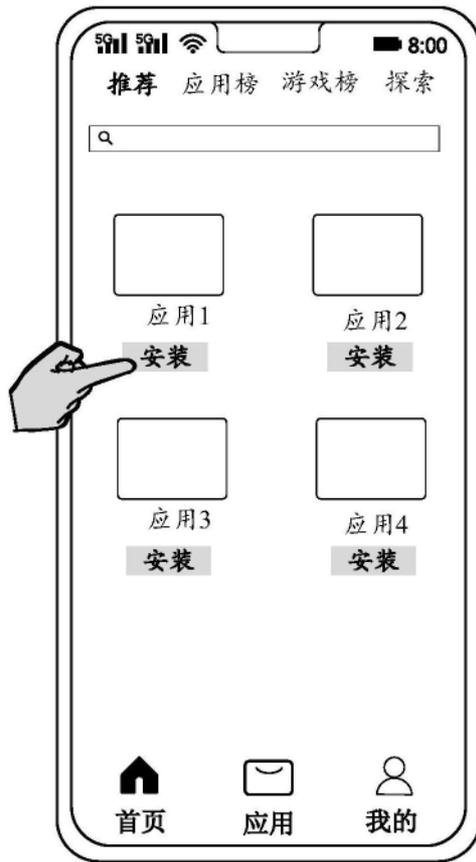


图11A

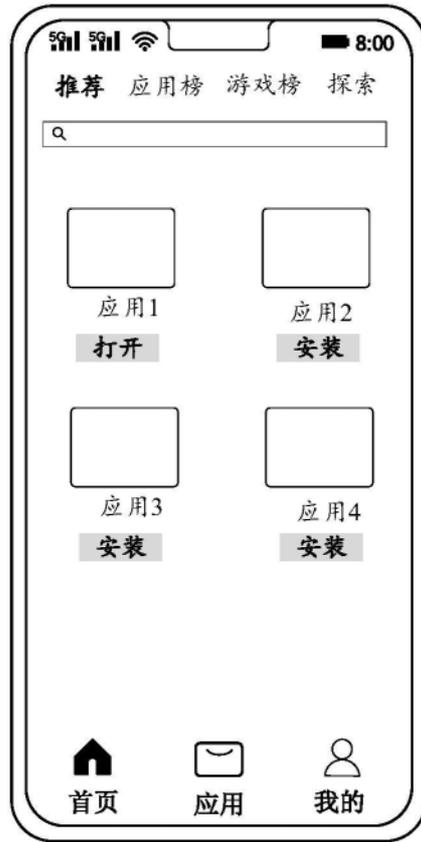


图11B

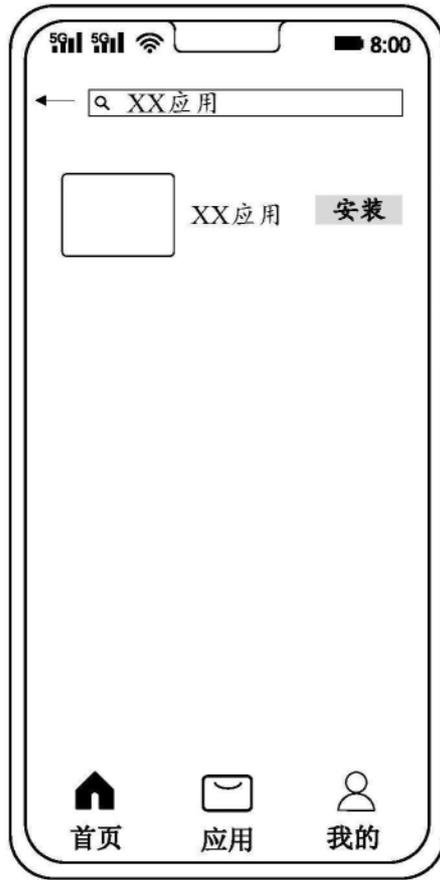


图11C



图11D

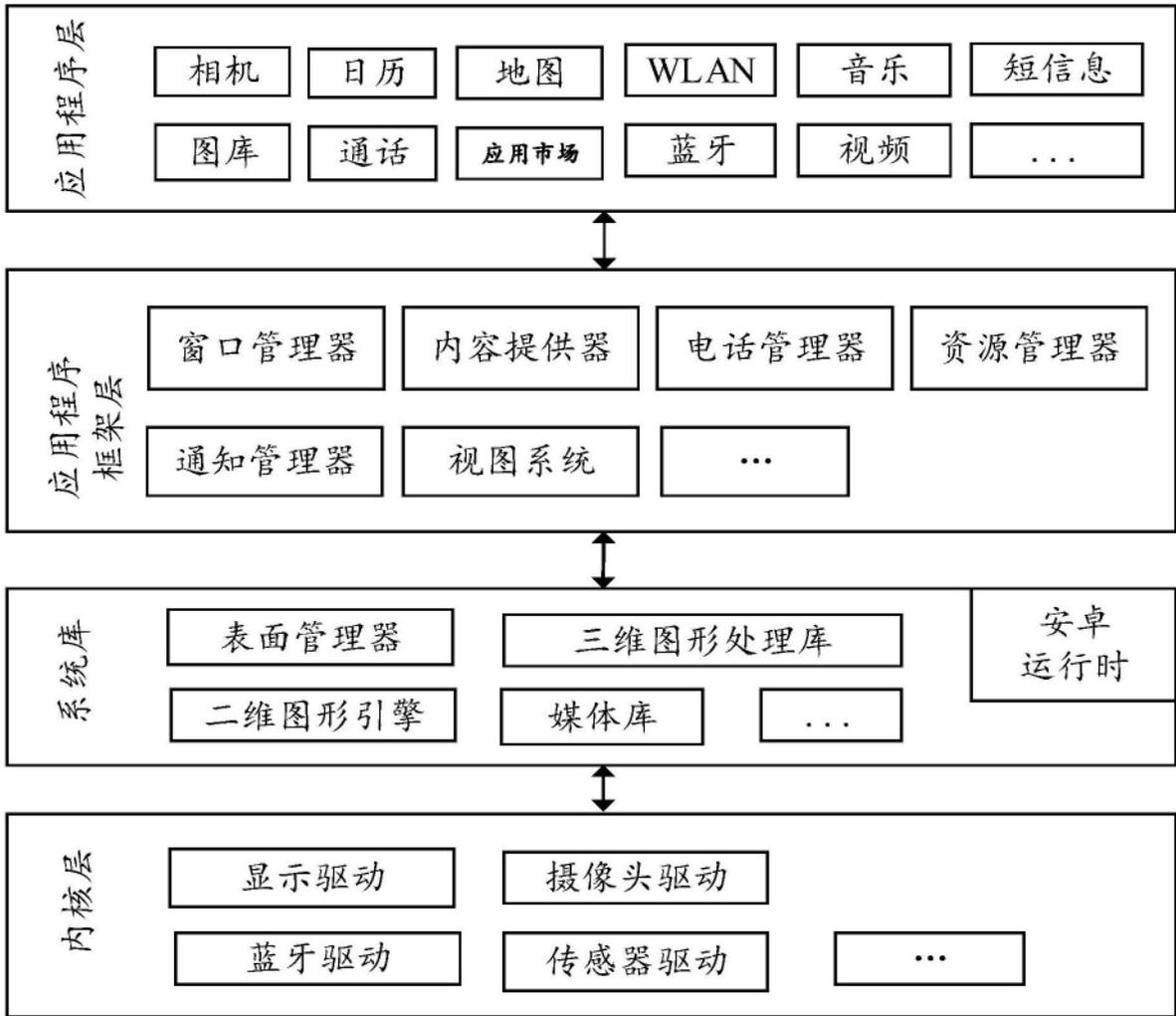


图12

电子设备100

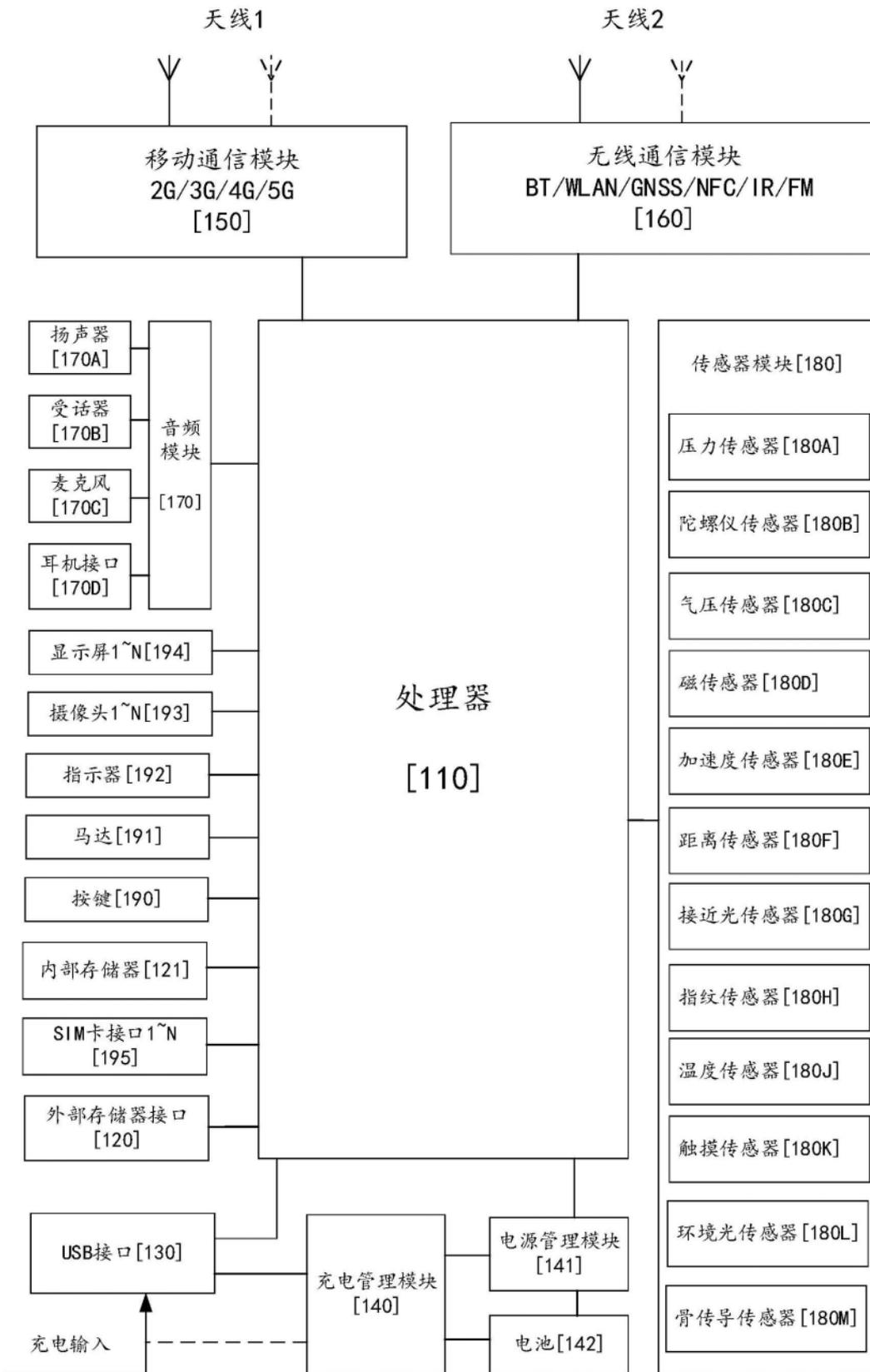


图13